Westinghouse

COMPUTER REFERENCE MANUAL

2500

WESTINGHOUSE 2500 INSTRUCTION SET

INSTRUCTION	MNEMONIC	FUNCTION CODE	TIMING (MICROSECONDS)
LOAD/STORE INSTRUCTIONS			
LOAD ACCUMULATOR	LDA	00101	1.4
LOAD B INDEX REGISTER	LDB	00001	1.4
LOAD C INDEX REGISTER	LDC	00010	1.4
LOAD EXTENDED ACCUMULATOR	LDE	00100	1.4
LOAD SHIFT DESCRIPTION REGISTER	LDG	00011	1.4
STORE ACCUMULATOR	STA	10101	1.4
STORE EXTENDED ACCUMULATOR	STE	10100	1.4
STORE ZERO	STZ	00111	1.4
JUMP INSTRUCTIONS			
CARRY JUMP	CJP	11111	1.0
NEGATIVE JUMP	NJP	10111	1.0
OVERFLOW JUMP	OJP	01111	1.0
POSITIVE JUMP	PJP	10110	1.0
UNCONDITIONAL JUMP	JMP	01110	1.0
ZERO JUMP	ZJP	11110	1.0
CONTROL INSTRUCTIONS			
CHANGE POST-INDEX DESIGNATORS	CDR	00110	1.0
CLEAR OVERFLOW DESIGNATOR	CDR	00110	1.0
* CHANGE LOCKOUT DESIGNATORS	CDR	00110	1.0
* RESET PROCESSOR INTERRUPT LOCKOUT	CDR	00110	1.0
INITIATE EXTENDED FUNCTION CODE	CDR	00110	1.0
ENTER STATUS	EST	11100	7.65
STORE STATUS	SST	11101	9.0
* STOP	STP	00000	0.9
ARITHMETIC INSTRUCTIONS			
ADD DOUBLE WORD TO ACCUMULATOR	ADA	11000	2.6
ADD WORD TO ACCUMULATOR	ADD	01000	1.3
DECREMENT LOCATION	DCR	01101	1.4
DIVIDE ACCUMULATOR	DIV	11011	26.0
INCREMENT LOCATION	INC	01100	1.4
MULTIPLY ACCUMULATOR	MPY	11010	18.0
SUBTRACT FROM DOUBLE-LENGTH ACCUMULATOR	SDA	11001	2.0
SUBTRACT FROM ACCUMULATOR	SUB	01001	1.3
LOGICAL INSTRUCTIONS			
AND WITH ACCUMULATOR	AND	01011	1.3
EXCLUSIVE OR WITH ACCUMULATOR	EOR	01010	1.3
SHIFT INSTRUCTIONS			
SHIFT WORD LEFT ARITHMETIC	SHF	10011	Note 1
SHIFT WORD LEFT CIRCULAR	SHF	10011	Note 1
SHIFT DOUBLE WORD LEFT ARITHMETIC	SHF	10011	Note 1
SHIFT DOUBLE WORD LEFT CIRCULAR	SHF	10011	Note 1
SHIFT WORD RIGHT ARITHMETIC	SHF	10011	Note 1
SHIFT WORD RIGHT CIRCULAR	SHF	10011	Note 1
SHIFT DOUBLE WORD RIGHT ARITHMETIC	SHF	10011	Note 1
SHIFT DOUBLE WORD RIGHT CIRCULAR	SHF	10011	Note 1
INPUT/OUTPUT INSTRUCTIONS			
* INPUT TO ACCUMULATOR	IOA	10001	0.9
* OUTPUT FROM ACCUMULATOR	IOA	10001	0.9
* INPUT TRANSFER REQUEST	ITR	10010	4.235
* OUTPUT TRANSFER REQUEST	OTR	10000	4.235

* NOT EXECUTABLE IN THE NON-PRIVILEGED MODE.

NOTE 1 SINGLE WORD SHIFT (2.00 + .74n) MICROSECONDS
DOUBLE WORD SHIFT (2.80 + 1.57n) MICROSECONDS

n = NUMBER OF SHIFTS

Westinghouse

PUBLICATION NO. 25REF-001A

PRICE: $4.75

SEPTEMBER 1971

COMPUTER REFERENCE MANUAL

2500

WESTINGHOUSE COMPUTER AND INSTRUMENTATION DIVISION
COMPUTER DEPARTMENT
1200 W. COLONIAL DRIVE, ORLANDO, FLA. 32804
AREA CODE (305) 843-7030

LIST OF EFFECTIVE PAGES

PAGE			ISSUE
i	THRU	vi	ORIGINAL
1-1	THRU	1-6	ORIGINAL
2-1	THRU	2-27	ORIGINAL
3-1	THRU	3-6	ORIGINAL
4-1	THRU	4-19	ORIGINAL
A-1	THRU	A-14	ORIGINAL

PRINTED IN U.S.A.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

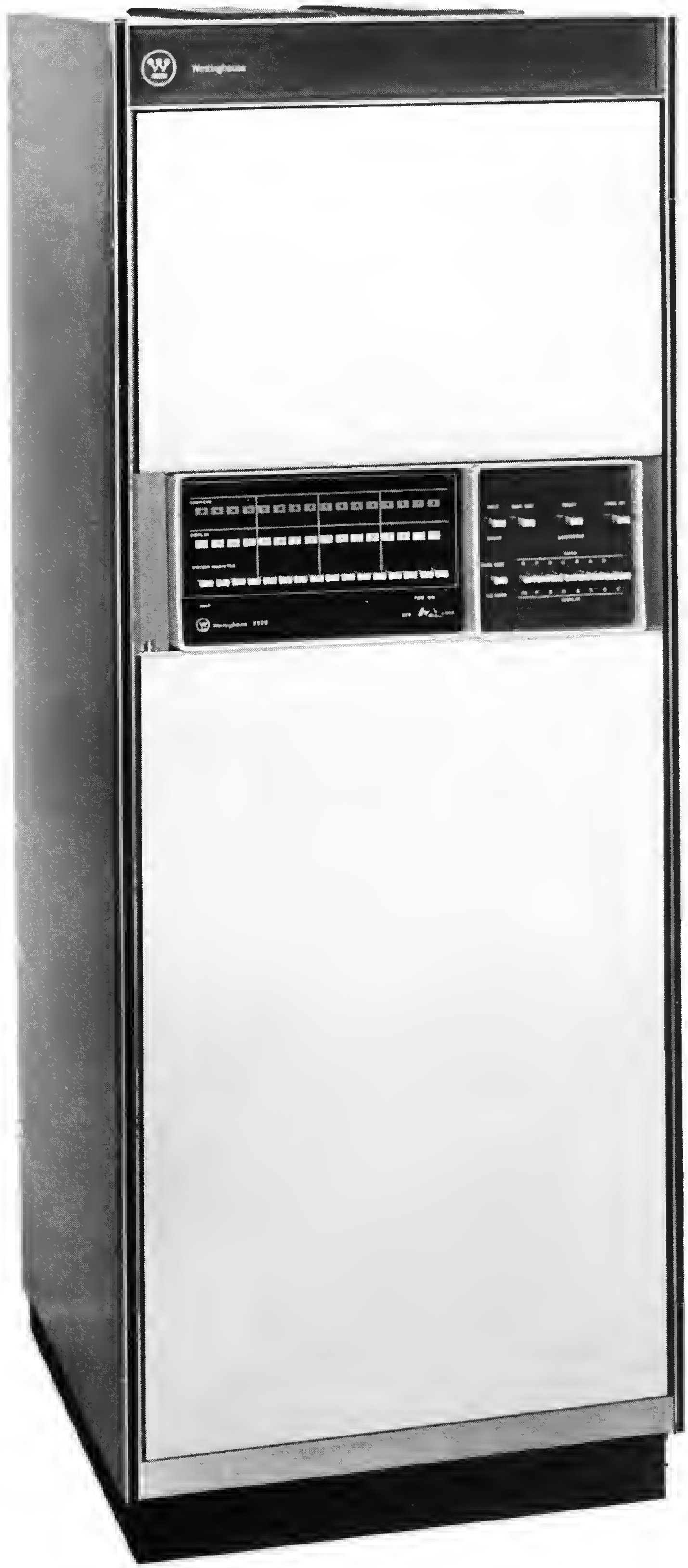
TABLE OF CONTENTS (Continued)

Title | Page

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

LIST OF TABLES



The Westinghouse 2500 Computer

SECTION 1

GENERAL DESCRIPTION

INTRODUCTION

The Westinghouse 2500 is a high-speed, 16-bit, general-purpose computer designed for use in a wide variety of advanced applications. Its speed and reliability make it ideal for the original equipment manufacturer (OEM), as well as for scientific, industrial, communications, and a variety of other applications.

Several standard hardware features are provided that are usually included only as options on other computers. Standard software packages are available to provide immediate capabilities beyond those normally associated with mini-computer systems.

HARDWARE FEATURES

Standard Features

The basic 2500 computer offers hardware features that include:

- Small Physical Size
- Large Memory Capacity
- Fast Memory Cycle Time
- Modular Construction
- Component Reliability
- Hardware Multiply and Divide
- Hardware Double Precision Add and Subtract
- Power Failure Detection
- Flexible I/O Design

Hardware Options

In addition to the standard features, the following options are available:

- Memory Parity
- Memory Protect
- Auto-Restart
- Hardware Bootstrap
- Memory Expansion
- External I/O Controller Chassis

SOFTWARE FEATURES

The Westinghouse 2500 Computer offers certain inherent features that provide for effective and efficient programming. These features include:

- Versatile Instruction Repertoire
- Extended Function Codes

- Multiple Addressing Modes
- Powerful Machine Control Instructions
- Sixteen Addressable Fast Access Registers

DESCRIPTION OF FEATURES

Brief descriptions of the features just presented are given in the following paragraphs.

Physical Size

The basic computer chassis, including up to 16K of core memory, is mounted in a RETMA 19-inch rack, taking 14 inches of vertical space and 21 inches of rack depth.

Memory Capacity

The memory capacity of the computer is 64K words. Core memory is provided in 4K (4096 words) increments. Up to 16K words of memory may be contained in the basic computer chassis.

Computer systems requiring core memory capacities in excess of 16K will require a configuration which includes:

- Basic computer chassis (first 16K of memory)
- Basic Computer power supply
- External memory chassis (can contain up to 48K of additional memory in increments of 4K)
- External memory power supply

Memory Cycle Time

Core memory cycle time is 750 nanoseconds. However, the first 16 addressable memory locations are integrated circuit, flip-flop registers with a cycle time of 450 nanoseconds.

Modular Construction

All circuits are constructed on printed circuit cards which plug into the basic chassis. Up to 34 printed circuit cards may be contained within the chassis. The maximum circuit card complement for the basic chassis is:

CARD TYPE	NO. OF CARDS
MEMORY SUBSYSTEM CARDS (ONE CONTROL CARD, ONE ADDITIONAL CARD FOR EACH 4K MODULE)	5
CENTRAL PROCESSOR CARDS	9
I/O SUBSYSTEM CARDS	4
FLOATING-POINT ARITHMETIC CARDS	5
PERIPHERAL INTERFACE CARDS	9
MEMORY PROTECT OPTION	1
HARDWARE BOOTSTRAP OPTION	1

Component Reliability

Maintenance requirements are minimized by the use of TTL (Transistor-Transistor-Logic) integrated circuit elements. These elements are grouped into functional circuits constructed on printed circuit cards.

Hardware Multiply and Divide

The basic computer contains hardware multiply and divide as standard equipment. The multiply instruction multiplies a 16-bit memory word by the 16-bit accumulator and leaves the product in the 32-bit extended accumulator (E- and A-registers). The divide instruction divides the 32-bit extended accumulator by the 16-bit memory word. The quotient is placed in the 16-bit accumulator (A-register) and the remainder is placed in the 16-bit E-register.

Double Precision Add and Subtract

Double precision arithmetic operations are provided as a standard feature. In double precision arithmetic a double-length word (32-bits) is added to or subtracted from the extended accumulator (E and A registers) and the result is placed in the extended accumulator.

Power Failure Detection

Power failure detection is an inherent feature of the 2500 power supply. When power fails and voltage begins to fall, a processor interrupt is generated. Thus, if an external power failure occurs, an orderly CPU shutdown is provided.

Flexible I/O Design

One common I/O bus provides interfacing capabilities for any Direct Memory Access (DMA), Buffered I/O, or Direct I/O device. Priorities are determined by the relative positions of the peripheral interface cards on the I/O bus.

The I/O subsystem provides an addressing capability for 128 controllers. Each controller may include logic for External Interrupts, Direct I/O data transfers, Buffered I/O data transfers or Direct Memory Access (DMA).

A maximum of 120 controllers may be used for External Interrupts, Direct I/O, and Direct Memory Access (DMA). A maximum of 62 controllers may be used for Buffered I/O.

External Interrupts——External Interrupts are handled by the I/O subsystem on a priority basis established by the positions of the interrupt circuits on a "daisy-chained" interrupt acknowledge line. Each External Interrupt may be identified by an I/O address (one of 120) or it may be grouped with other External Interrupts (16 are identified by a single I/O address) and handled by a data word transfer.

Direct I/O——The Direct I/O circuits on a controller are used for status and control word transfers between the accumulator and the DMA or Buffered I/O interfaces or they may be used for direct data transfers between the accumulator and an external device.

Buffered I/O Controllers——The Buffered I/O controllers perform block transfers of data to and from memory under control of out-of-sequence instructions. The out-of-sequence instruction is issued in response to a Service Request Interrupt generated by the Buffered I/O controller. These Service Request Interrupts are all lower in priority than External

Interrupts and are arranged in a priority structure based on the relative position of each controller card on the I/O bus.

Direct Memory Access (DMA)—Direct Memory Access controllers perform data transfers between memory and the I/O device through the I/O subsystem. Transfers are made on a memory-cycle-steal basis in response to DMA requests from the DMA controllers.

Memory Parity

The memory parity option is provided in the memory subsystem. The subsystem generates odd parity on a word basis. When a parity error is detected, the subsystem generates a Processor Interrupt. If an out of core address is accessed, a parity error is also generated.

Memory Protect

The memory protect option is implemented in two ways: non-privileged instruction mode of operation, and memory write lockout. If the memory protect option is not installed, all programs are run in the privileged mode (all instructions are executed).

Auto-Restart

When the processor interrupt occurs in response to a power failure, an auto-restart circuit may be armed by the program as part of a normal power shutdown sequence. When power returns, the auto-restart circuit generates the START signal for the processor.

Hardware Bootstrap

The optional hardware bootstrap is instituted as a diode ROM (Read Only Memory). The contents of the ROM are automatically loaded into core memory by pressing the BOOTSTRAP switch on the control panel.

External Memory

Memory capacities above 16K require an external, rack-mounted, memory chassis which provides for an additional 48K of memory (may be in same cabinet with CPU).

External I/O Controller Chassis

Systems requiring more than nine peripheral controller cards must be supplied with the extended peripheral controller chassis. When the external chassis is used, the basic CPU is equipped with the peripheral expander card.

Instruction Set

Standard Instructions—The standard 44 instructions for the Westinghouse 2500 include hardware multiply and divide, double precision add and subtract, increment or decrement a location, exclusive OR and logical AND. The complete instruction set is summarized on the inside front cover of this manual and there is a description of each instruction in Section 4.

Extended Function Code—The Change Designator Register (CDR) instruction is further defined to provide additional instructions (extended function codes) for the operator's panel read and display functions. The extended function codes are described in Section 4 under the CDR instruction.

Multiple Addressing Modes

All memory reference instructions can specify direct (absolute or relative) or indirect (absolute or relative) addressing. Indexing operations are provided by two index registers which may be used for pre-indexing and post-indexing operations.

Machine Control Instructions

Four machine control instructions are provided: Enter Status (EST), Store Status (SST), Change Designator Register (CDR), and Stop (STP). These instructions facilitate entry and exit from subroutines, control the setting of the designator register, and stop the CPU for manual control.

Fast Access Registers

The first 16 addresses are fast access flip-flop registers. The first six of these registers are working registers (Accumulator, E-Register, P- Register, etc.) and the next ten are general purpose registers which may be used by the program whenever a fast access register is required.

GENERAL SPECIFICATIONS

The general specifications for the Westinghouse 2500 are shown in Table 1-1.

Table 1-1. General Specifications

Specification	Characteristic
Type	Sixteen bit, general purpose stored program digital computer.
Core Memory	4096 words standard, expandable to 64K in increments of 4096.
Fast Access Registers (FAR)	The first 16 addresses are 16-bit fast access flip-flop registers. (Fast Access Registers)
Memory Cycle Time:	450 nanoseconds - Fast Access Registers 750 nanoseconds - Core Memory
Addressing	Direct and indirect, absolute and relative, pre-indexing and post-indexing.
Arithmetic	Parallel, binary, fixed point, two's complement. Double precision add/subtract. Hardware multiply and divide. Optional floating point hardware.
Instructions	8 Load/Store, 6 Jump, 8 Control, 8 Arithmetic, 2 Logical, 8 Shift, 4 Input/Output. Total = 44
Input/Output	The I/O subsystem provides interfacing for any mix of External Interrupt, Direct I/O, DMA or Buffered I/O controllers providing total addresses do not exceed 120 and Buffered I/O controllers do not exceed 62.
Packaging	Provided in 19 inch RETMA rack mounted configuration.
Power	120 Vac, 60 Hz, single phase.

SECTION 2

FUNCTIONAL DESCRIPTION

INTRODUCTION

This section describes the computer from a functional viewpoint to provide the user with a general knowledge of the processor operation and word formats.

The Westinghouse 2500 is a digital computer which uses, as its basic information unit, a 16-bit binary word constructed as shown in the following diagram:

BIT	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BINARY	0	0	1	1	1	1	1	1	0	0	1	0	1	0	1	0
HEXADECIMAL	3				F				2				A			

BASIC WORD

Using this format, bit position 15 represents the most significant portion of the word and bit position 0, the least significant. This basic informational scheme is reflected in the operational registers and the adder.

This basic 16-bit structure makes it convenient to use hexadecimal notation to express the binary information processed by the computer. Using hexadecimal notation, a 16-bit binary number requires only four digits. Table 2-1 shows the hexadecimal and binary equivalents for decimal numbers 0 through 15.

Table 2-1. Decimal/Binary/Hexadecimal Equivalents

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Throughout this manual, hexadecimal notation is represented by a string of hexadecimal digits enclosed in single quotation marks and preceded by the letter X. For example, the hexadecimal notation for the decimal number 16,170 is written as X'3F2A'. More information on hexadecimal notation, including conversion tables, is given in Appendix A.

A functional block diagram of the computer showing the basic operational elements and data flow through these elements is shown in Figure 2-1. For descriptive purposes, the computer may be divided into four functional groups: memory, arithmetic and control, extended system port, and the input/output subsystem.

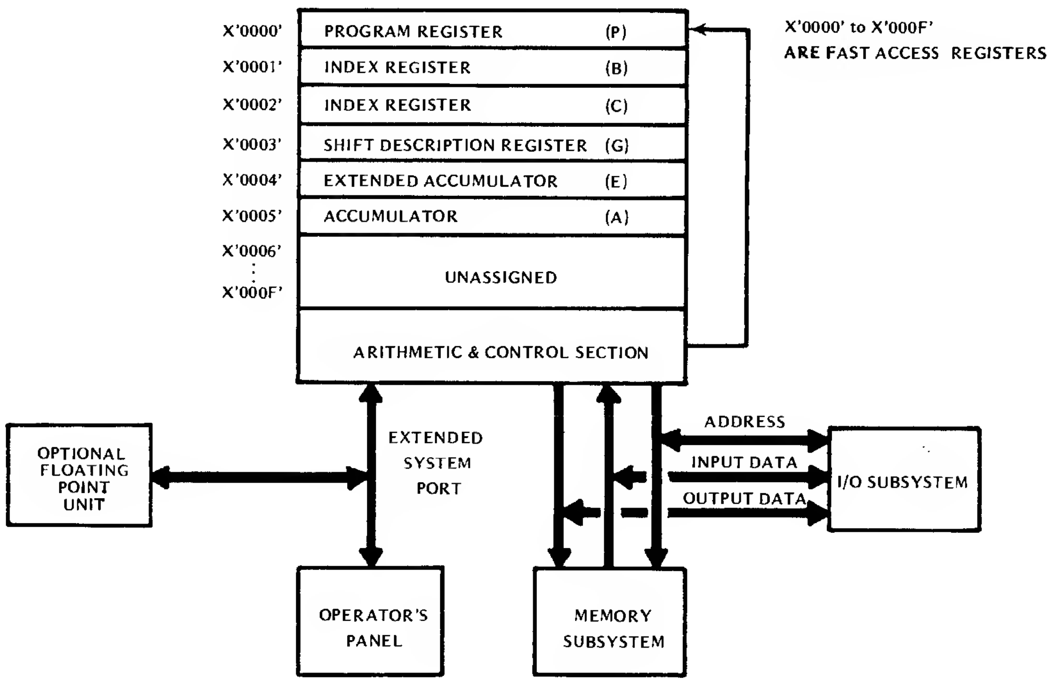


Figure 2-1. Block Diagram — Westinghouse 2500 Computer

MEMORY

The first 16 memory locations of the computer are high speed, flip-flop registers. These registers are designated Fast Access Register (FAR) and are physically separated from core memory. The remainder of memory is random access core which has a maximum capacity of 64K (K=1024) words. Core memory size is optional and is supplied in 4096 word increments up to 16,384 words internally, and up to 65,536 words using the external memory rack and power supply. Certain locations in memory are reserved as defined in subsequent paragraphs.

Fast Access Registers

The Fast Access Registers (FAR) are controlled by the arithmetic and control section in response to memory addressing or an instruction execution sequence. The FAR data bus is internal to the arthmetic and control section where it is used by the operational registers.

The first six Fast Access Registers are program working registers and are not used for general computation. The remaining ten locations are available for any function where the high-speed access time is desirable. The first six registers are described in the following paragraphs.

Program (P) Register——The program register, addressable as location X'0000', contains the address of the instruction being executed. At the initiation of an instruction execution, the P-register contains a number which is one less than the address of the instruction to be executed. If, for example, the instruction located at 01F3 is to be executed, the P-register contains 01F2 at the start of the instruction sequence. The P-register is then incremented to 01F3 and the instruction is fetched from memory.

Incrementing the P-register before the instruction is executed means that the operand for all jump, SST, and EST instructions that are coded directly in machine language must be specified as operand-1 so that control will be transferred correctly when the program is executed.

NOTE

When using one of the SYMBAL processors, the operand is automatically decremented for these instructions when direct addressing is used.

Index (B & C) Registers——The B and C index registers are addressable as memory locations X'0001' and X'0002', respectively. The index registers are also accessed by the arithmetic and control section during address calculations for pre- and post-indexing operations.

Shift Description (G) Register——The G-register is addressable as memory location X'0003'. It is loaded by the program, prior to a shift instruction, with a shift-description word which specifies the shift type and count. The G-register is read by the arithmetic and control section during the execution of a shift instruction. See LDG instruction in Section 4 for the shift-description word format.

Accumulator (A) Register——The A-register is addressable as memory location X'0005'. It stores the results of arithmetic operations, transfers data words to the I/O subsystem and Direct I/O Controllers, and serves as temporary storage for general program manipulations.

Extended Accumulator (E) Register——The E-register is addressable as memory location X'0004'. It augments the size of the accumulator in arithmetic operations which require a double-word length. In double precision operations, the E- register contains the most significant half of the double word and the A- register contains the least significant half.

Reserved Memory Locations

In addition to the FAR registers just defined, there are certain locations in core which are reserved for processor use. These locations are defined in Table 2-2.

ARITHMETIC AND CONTROL

A functional block diagram of the arithmetic and control section registers, and their transfer

paths, is shown in Figure 2-2. The registers shown in this diagram, along with the dedicated registers in FAR previously defined, are used to perform instruction executions.

Table 2-2. Reserved Memory Locations

Address	Use
X'0000'	P (Program) Register
X'0001'	B Index Register
X'0002'	C Index Register
X'0003'	G (Shift Description) Register
X'0004'	E (Extended Accumulator) Register
X'0005'	A (Accumulator) Register
X'0100'	External Interrupt
X'0101'	Processor Interrupt (Memory Parity & Power Fail)
X'0102'	Buffer Overflow Service Request Interrupt (SRI)
X'0103'	Processor Interrupt (Memory Write & Instruction Violation)
X'0104'	Buffered I/O SRI
⋮	⋮ ⋮ ⋮
X'017F'	Buffered I/O SRI

S-Register

The S-register is a 16-bit register, primarily used to store addresses during memory and I/O sequences. Inputs to the register are:

1. The output of the P-register (incremented by one).

2. The Arithmetic Logic Unit

The output of the P-register is incremented by one and loaded into the S-register to fetch an instruction. The output of the Adder is loaded into the S-register as a result of operand address calculations. Outputs of the S-register address memory, drive the indicator lights on the operator's panel (ADDRESS), restore the incremented program count to the P-register, and provide true and complementary inputs to the X-register.

Z-Register

The Z-register serves as a buffer for the Extended System Port data bus, the input data bus, the D-register and the FAR data bus. It is also used to shift data during shift operations.

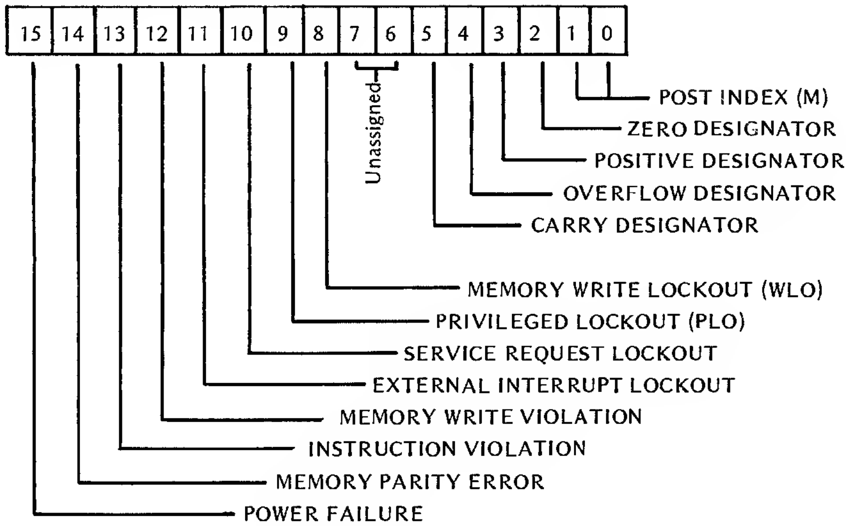
X-Register

The X-register is used for temporary storage of data from memory or the I/O subsystem to free the Z-register for another access. The X-register is also used as a buffer register for FAR when FAR and core memory are accessed simultaneously.

Designator (D) Register

The Designator (D) register is a 16-bit flip-flop register which contains binary information

showing the current status of the machine. Bit assignments for the D-register are shown below and are described in Table 2-3.



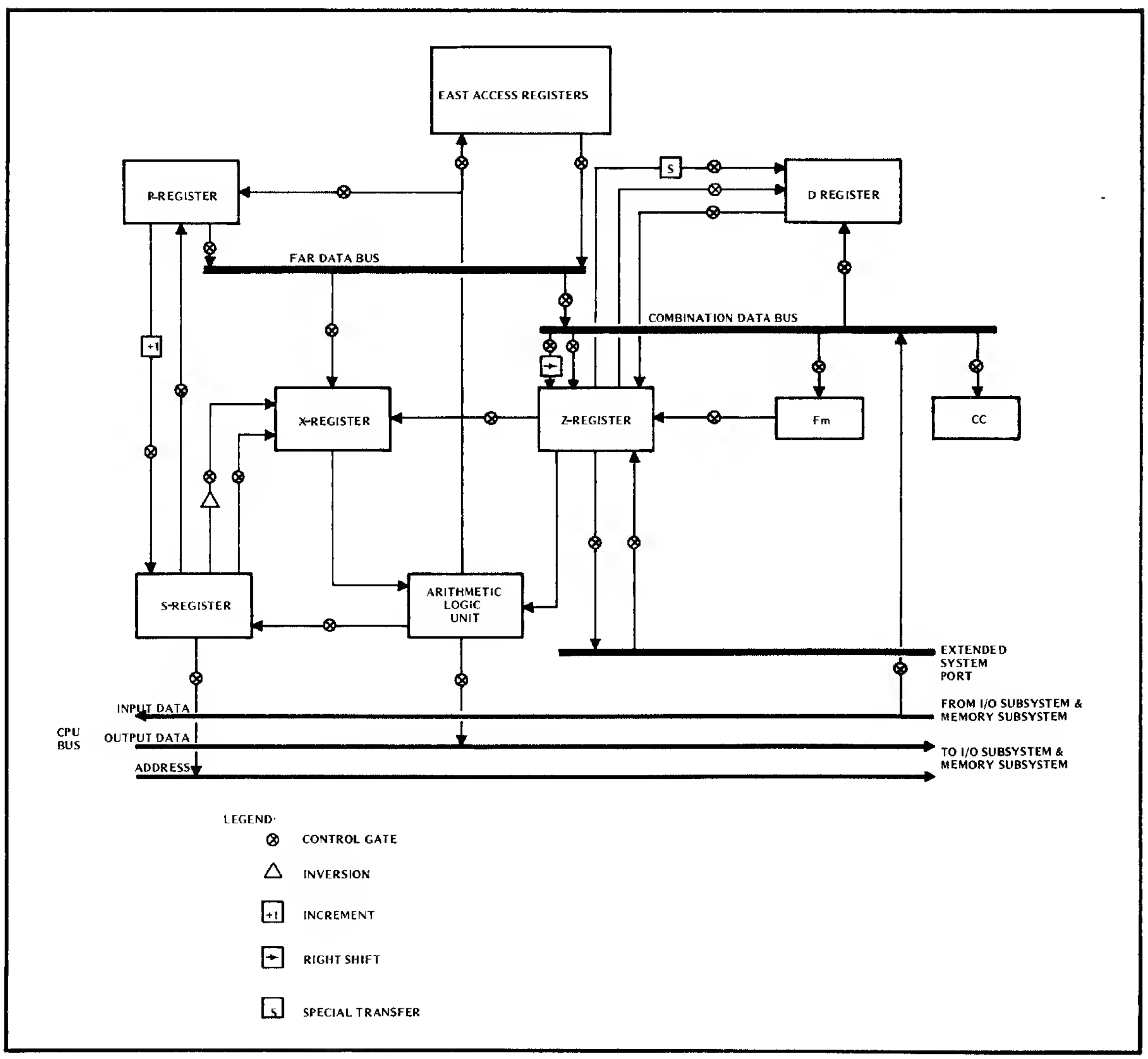


Figure 2-2. Functional Block Diagram — Arithmetic and Control

Table 2-3. Designator Register Bits

BITS	NAME	HOW SET	HOW RESET	ACTION
0 1	Post Index (M)	CDR, EST* execution.	CDR, EST* execution.	Causes indexing as per Table 2-5.
2	Zero Designator	Result of arithmetic operation = 0, EST* execution.	Arithmetic operation = 0, EST* execution.	ZJP (zero jump) instruction interrogates zero designator and makes decision.
3	Positive Designator	Arithmetic operation, set when bit 15 of high order word = 0, EST* execution.	Negative result, arithmetic operation, EST* execution.	NJP (negative jump) and PJP (positive jump) instructions interrogate positive designator and make decision.
4	Overflow Designator	EST* execution, set if the sign bit changes during arithmetic operation.	CDR execution, EST* execution.	OJP (overflow jump) instruction interrogates overflow designator and makes decision.
5	Carry Designator	EST* execution, set by carry during arithmetic or shift operations.	Arithmetic operation, EST* execution, shift operation.	CJP (carry jump) instruction interrogates carry designator and makes decision.
6 7	Unassigned			
8	Memory Write Lockout (WLO)	CDR execution (privileged), EST* execution (non-privileged)	CDR execution (privileged), EST* execution (non-privileged)	Initiates Memory Write Protect. When set, an instruction execution outside of memory write limits causes a processor interrupt.
9	Privileged Instruction Lockout (PLO)	CDR execution (privileged), EST* execution (non-privileged)	CDR execution (privileged), EST* (non-privileged)	Initiates Privileged Instruction Mode. When set, the execution of CDR, IOA, ITR, OTR, or STP can be executed only when PLO is RESET. (Bit 9 = 0). Attempted execution of these instructions when bit 9 is set causes a Processor Interrupt.
10	Service Request Lockout	CDR, EST* execution, Processor Interrupt.	CDR, EST* execution.	When set, prevents Service Request Interrupts from being serviced.
11	External Interrupt Lockout	CDR, EST* execution, Processor Interrupt.	CDR, EST* execution.	When set, prevents External Interrupts from being serviced.
12	Memory Write Violation	Out of limit memory write.	SST	When set, causes a Processor Interrupt to location X'103'.
13	Instruction Violation	Execution of: 1. Un-implemented extended op-code. 2. Non-privileged CPU mode and privileged instruction. 3. ITR/OTR instruction outside of SRI location.	SST	When set, causes a Processor Interrupt to location X'103'.
14	Memory Parity Error	Memory Parity Error during memory read.	SST	When set, causes a Processor Interrupt to location X'101'.
15	Power Failure	Detection of an ac power interruption.	SST	When set, causes a Processor Interrupt to location X'101'.

The contents of the D-register may be stored in memory by the SST instruction, loaded from memory by the EST instruction, and altered by the CDR instruction (See Section 4). In machines equipped with Privileged Instruction Lockout (see Memory Protect), reading and writing the D-Register differs slightly depending on the state of the Privileged Lockout designator (D-Register, bit 9). In machines not equipped with Privileged Instruction Lockout, the CPU operates in the privileged mode. The following summaries define the actions taken during a designator read and write in all modes of machine operation.

Designator Write——Bits 0 through 7 may be written by arithmetic operations, CDR instruction (m=0), and EST instructions regardless of the state of the Privileged Instruction Lockout designator (bit 9). Bits 8 through 11 may be written by CDR (m=1 or 3) instructions or EST instructions only when the Privileged Instruction Lockout designator (bit 9) is ZERO (privileged mode of operation). Bits 12 through 15 cannot be set by the program but are set when the appropriate violation occurs and cleared by an out-of-sequence SST in response to the appropriate Processor-Interrupt.

Designator Read——Bits 0 through 7 may be read by SST and conditional jump instructions regardless of the state of the Privileged Instruction Lockout (bit 9). Bits 8 through 11 may be read by an SST instruction when in the Privileged Mode or when executed as an out-of-sequence instruction.

Control Counter (CC) Register

The CC-register is used as a secondary sequencer during the execution of Shift, Multiply, Divide, Enter Status, and Store Status instructions. The upper three bits of the register are used to indicate the type of shift operation to be performed.

Arithmetic Logic Unit (ALU)

The ALU performs the following arithmetic and logical functions from inputs provided by the X- and Z-registers:

EXCLUSIVE OR
AND
ADD
SUBTRACT

The ALU output is selectively gated to the I/O Subsystem and Memory Subsystem (via the CPU Bus), to the S-register, and to the Fast Access Registers (FAR).

The following list summarizes the operations performed by the 2500:

Instruction	Operation	Designators Affected
Add	A + W → A	Zero (D2)
Add double word	"E+A" + "W+(W+1)" → "E+A"	Positive (D3)
Subtract	A - W → A	Overflow (D4)
Subtract double word	"E+A" - "W+(W+1)" → "E+A"	Carry (D5)
Increment	W + 1 → W	
Decrement	W - 1 → W	
Multiply	A x W → "E+A"	
Divide	"E+A" ÷ W → A=Quotient E=Remainder	

Instruction	Operation	Designators Affected
Shift	$A \times [\pm S_2] \rightarrow A$ $“E+A” \times [\pm S_2] \rightarrow “E+A”$	Zero (D2) Positive (D3) Overflow (D4) Carry (D5)
AND	$A \cdot W \rightarrow A$	Zero designator, positive designator only
Exclusive OR	$A + W \rightarrow A$	

Legend:

A = contents of the accumulator
E = contents of the E-register
W = contents of designated memory location
S_2 = number of binary shifts
“E + A” = concatenation of E + A

Fm-Register

The Fm-register is an 8-bit register used to store the function code (f) and address (m) mode bits of an instruction for execution decode.

EXTENDED SYSTEM PORT

The Extended System Port (ESP) provides data transfer and control for the Operator's Control Panel. The ESP data bus is interfaced to the Arithmetic and Control section through the Z-register.

INPUT/OUTPUT SUBSYSTEM

A functional block diagram of the input/output subsystem is shown in Figure 2-3. The subsystem provides control for all data transfer between the peripheral devices and the CPU or memory. In addition, the subsystem handles all External Interrupts and Buffered I/O Service Request Interrupts on a priority basis and supplies interrupt numbers to the CPU. In general, the subsystem provides control for Direct I/O transfers, Buffered I/O transfers, Direct Memory Access transfers and External Interrupts.

Direct I/O Transfers

Direct I/O transfers are used to perform three basic functions for peripheral devices: data transfers, control word transfers, and status word transfers. All transfers are initiated by the execution of an IOA instruction in the CPU. IOA instructions may specify direct or indirect addressing (see Figure 2-4). In the direct mode, bits 0 through 6 of the instruction specify a controller address and bit 7 specifies direction of transfer. In the indirect mode, the operand contains the channel address in bits 0 through 6, the direction indicator in bit 7, and the function code or device address in bits 8 through 15. The I/O subsystem transfers the I/O controller address and direction bit (or the full operand word) to the device interface over the address lines and transfers the data word over the data lines. If a device does not respond within five microseconds after it is addressed by the I/O subsystem, an external interrupt is generated and the type of interrupt is stored at device address X'7D' (last direct I/O device time-out).

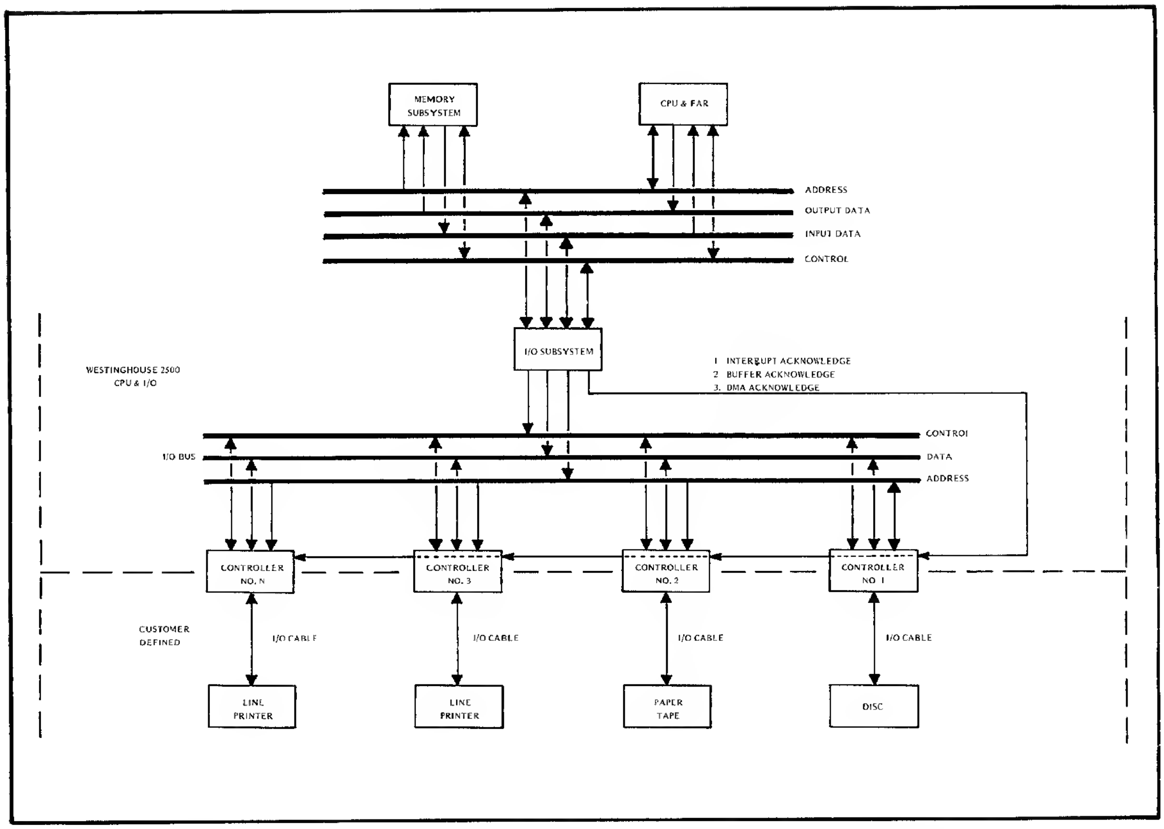


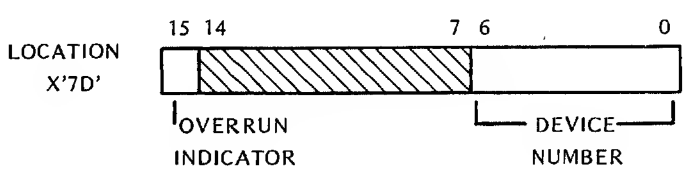
Figure 2-3. Functional Block Diagram — Input/Output Subsystem

Data Transfers are made between the A-register and the direct I/O interfaces by IOA instructions. An External Interrupt may be used by the direct I/O interface as a transfer completion signal.

Control Word Transfers are performed in the same manner as data transfers but are used to control or initiate data transfers between the CPU and buffered I/O or DMA devices. Completion interrupts may or may not be used in response to control word transfers.

Status Word Transfers are direct input data transfers used to transmit device status information. These transfers may be used to interrogate device status prior to data transfers or upon detection of an error condition.

Direct I/O Timeout——If a device does not respond within five microseconds after it is addressed by the I/O subsystem, an external interrupt is generated and the type of interrupt is stored at device address X'7D'. If more than one device times-out before X'7D' is read, the overrun indicator (bit 15) is set. The word format for location X'7D' is as follows:



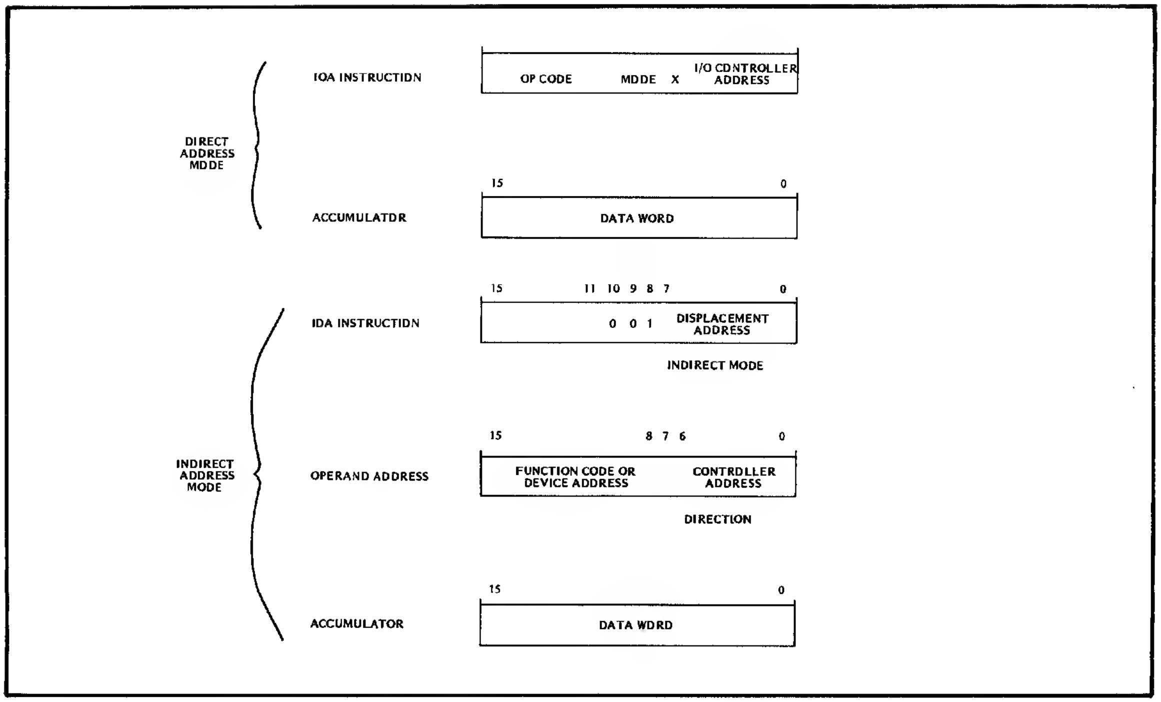


Figure 2-4. Direct I/O Instruction

Direct I/O Interrupt—–If a direct I/O device causes an External Interrupt, device address X'7F' will have bit 15 set and bits 6 through 0 will contain address X'7F'. NOTE: Device address X'7F' can contain notification of Buffered and DMA interrupts also. Refer to the section on External Interrupts for additional information.

Buffered I/O Transfers

The Buffered I/O transfers are usually initiated by an IOA instruction which transfers the control information to the device controller by means of the direct input/output interface. Thereafter, the controller generates Service Request Interrupts (SRI) as data is required or is available.

When the buffered I/O device is ready to transfer data, it generates a SRI. When the computer honors this request, it halts the running program, transfers a single data word, and resumes the program as soon as the transfer is complete. This method of executing out-of-sequence instructions transfers data efficiently and is transparent to the running program.

Buffered I/O-SRI Number—–Each buffered device is assigned a Service Request Interrupt (SRI) number which is usually the same as the device number. For example, the teletype is assigned device number X'010' and SRI number X'010'. Additionally, each device's Service Request Interrupt (SRI) is hardware-assigned two memory locations which are defined as X'100 + (2 x SRI No.) and X'101' + (2 x SRI No.). The two memory locations contain an ITR or an OTR instruction and the buffer size as shown below:

Location No. 1

X'100 + (2 x SRI No.)	contains an OTR (output) or ITR (input) instruction.

Location No. 2

X'101' + (2 x SRI No.)	contains a number which is the two's complement of the buffer size (negative index count).

The special SRI operand address calculation for the first word is performed as follows:

1. X'100' + instruction word's displacement = indirect address.
2. Contents of indirect address + second word's negative index count = operand address.

Buffered Output Operation—–In this example, it is assumed that the Teletype I/O controller is wired to SRI number 10 and the contents of the 3-word buffer X'5000' - '5002' are to be transferred to it. The programmer has loaded the 3-word output buffer with data in the format required by the I/O device (ASCII).

SRI Number 10 Memory Definition—–The memory locations assigned to Service Request Interrupt number 10 are X'100' + (2 x 10) = X'120' and X'101' + (2 x 10) = X'121'.

Program Initialization—– Location X'180' contains the output buffer end address +1 (X'5003'). The SRI locations contain the following:

ADDRESS	CONTENTS
X'120'	X'8080' (OTR INSTRUCTION WORD)
X'121'	X'FFFD' (TWO'S COMPLEMENT OF THE BUFFER SIZE)
X'180'	X'5003' (OUTPUT BUFFER END ADDRESS + 1)

SRI Operation—–The SRI operation is as follows:

1. The request mode of operation is initiated for Service Request Interrupt number 10.
2. The instruction in location X'120' is obtained and the address is calculated as follows:
   - X'100 is added to the instruction word displacement field; result: X'180'.
   - The contents of location X'0180' are obtained; result: X'5003'.
   - The contents of location X'0180' are added to the index count and the result (first cycle, X'5003-3 = X'5000') is the operand address.
   - The negative index count in location X'121' is incremented. (First transfer, to -2). When the incremented index count reaches zero, the Buffer Overflow Service Request Interrupt (No. 1) is generated.

3. The OTR is executed using the calculated operand address. Contents of location X'5000' are transferred to the I/O device.

4. The SRI number is used to address the channel, thus, the data is gated into the addressed I/O device.

After the OTR execution, the running program is immediately resumed. When the I/O device data transfer function is complete, the Service Request Interrupt operation is repeated. This cycling is continued until all the words in the buffer have been transferred. When the buffer is empty, the Buffer Overflow Interrupt is generated and cycling is stopped.

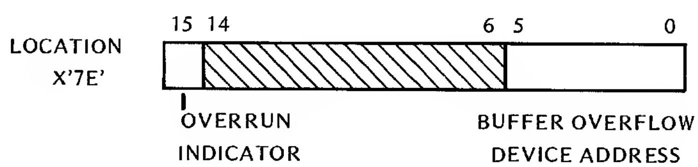
Buffered Input Operation—Buffered input channel operation is similar to output. Assume the same parameters as given in the output example, except that the 3-word buffer at X'5000' is to be loaded with data from the I/O device. The entire operation is then the same as for the output example with the following exceptions:

1. Location X'120' will contain ITR 80 (X'9080').

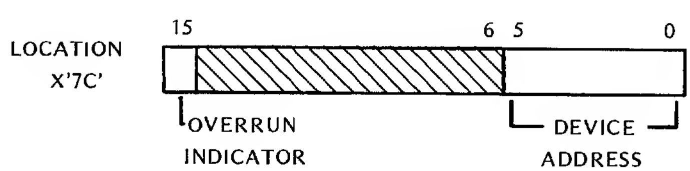
2. When the ITR is executed using the calculated operand address, the data is gated from the I/O device and stored directly in the buffer location.

Buffer Overflow—The buffer overflow interrupt is generated when a buffer index count goes to zero and signifies that a buffered transfer is complete. The buffer overflow interrupt location (X'102') is a special one- word SRI location which generally contains an SST instruction. The indirect operand address for this instruction is calculated by adding X'100' to the instruction's displacement field.

Standard device address X'7E' contains the address of the last buffered device which caused an overflow condition. If more than one overflow occurs before X'7E' is read, the overrun indicator (bit 15) is set. The buffer device overflow word format is shown below:



Buffered I/O Timeout—If a buffered I/O device does not respond to an ITR or OTR instruction within five microseconds after it is addressed by the I/O subsystem, an external interrupt is generated. Standard device address X'7C' contains the address of the last buffered I/O device that timed out. If more than one device times out before X'7C' is read, the overrun indicator (bit 15) is set. The buffer device timeout word format is shown below:



Other Instructions in a Buffered Location

Occasionally, instructions other than ITR and OTR instructions are used in a buffered location. The execution is the same as if the instruction were in any other location with the exceptions noted below and the address calculation is the same as that used by the ITR/OTR instruction.

Address Calculation:

1. Add X'100' to the displacement field.
2. Use the resulting address as an indirect address.
3. Add the index count contained in the second buffered location to the indirect address to obtain the operand address.
4. If the index count is negative, it is incremented each time the device generates an SRI until the count goes from -1 to 0. At this time the Buffer Overflow Interrupt is generated.
5. If the index count is zero or positive, the second word is added to the contents of the indirect address as described, but the count is not incremented.

The IOA and CDR instructions are exceptions to the above:

If an IOA instruction is placed in a buffered location, and the program attempts to execute it, a NO-OP is performed and an Instruction Violation Processor Interrupt is generated. If a CDR instruction is placed in a buffered location, the instruction is executed but address calculations are ignored and the index count is not incremented.

A jump instruction executed as an out-of-sequence instruction in a buffered location can cause the machine's status to be lost when control is transferred (use the SST to Preserve Status and to transfer control).

Direct Memory Access (DMA)

Direct memory access transfers are made between memory and the DMA device interface on a cycle-steal basis. These transfers are made automatically by the input/output system in response to priority requests from the DMA devices.

A DMA transfer is usually initiated by IOA instructions which provide necessary device control instructions, beginning memory address, transfer count, and a start command. Thereafter, data transmissions are made directly to or from memory through the I/O subsystem on a cycle-steal basis.

DMA Timeout––If a DMA device does not respond with a start signal to the I/O subsystem within five microseconds after being addressed, an external interrupt is generated. Memory location X'7F' then contains a word in which bit 13 is set to signify that there has been a DMA timeout. Additionally, the I/O subsystem forces a memory read cycle so as to protect the contents of memory from an erronous write cycle.

External Interrupt Processing

External Interrupts are acknowledged and processed on a priority basis, determined by the position of the interrupt controller on the I/O bus. Each controller can raise one common request line to the I/O subsystem. When this signal is raised and any current transfer is complete, the I/O subsystem acknowledges the External Interrupt by sending a signal through a "daisy-chain" line connecting all interrupt controllers. The first controller in the chain to have an external interrupt request terminates the acknowledge signal and sends its interrupt number to the I/O subsystem. The I/O subsystem then generates the External Interrupt for the CPU and holds the interrupt number in a register.

When the CPU is ready to process the External Interrupt, it executes an out-of-sequence instruction stored at location X'100'. This instruction should then transfer to a subroutine which contains an IOA instruction to read device address X'7F' and transfer the interrupt number (device address) to the A-register.

For additional information refer to the section on interrupts.

Standard Device Address Assignments

The standard addresses for devices are listed in Table 2-4. Normally, devices which use some combination of direct I/O, buffered I/O, and External Interrupt use the same device address throughout. For example: the line printer uses buffer channel X'0C', direct channel X'0C', and external interrupt X'0C'. Direct Memory Access (DMA) peripherals communicate with the CPU via IOA instructions.

MEMORY REFERENCE INSTRUCTIONS

General

Memory reference instructions are capable of addressing any memory location within the 64K capacity of the Westinghouse 2500. The methods used in addressing are diagrammed in Figure 2-5 which shows the word format and the relationship of the three fields (function, mode, and displacement address) to the hexadecimal printout or display. Note that the second hexadecimal character determines the addressing mode and that it ranges from 0 through 7 or 8 through F for each memory reference instruction. The second character is further defined by bits 0 and 1 of the Designator (D) register when pre-index indirect operations are specified.

Function (F) Field

The F-field is comprised of five bits (15 through 11) which specify the instruction to be executed. The contents of this field are expressed in hexadecimal notation and are referred to as the Op Code. The most significant hex digit of the op code is derived from bit 11 (the most significant bit of the next four bits) and is, therefore, always 0 or 8. For programming purposes, each op code is assigned a three-letter mnemonic symbol which can be read and interpreted by the Symbolic Assembler.

Address Mode (m) Field

The m field is comprised of three bits (bits 10, 9 and 8) which specify the address mode used to calculate the operand address. The address modes are defined in Table 2-5. If pre-indexing operations are specified for indirect addressing, then bits 1 and 0 of the Designator register are interrogated for the type of modification to be made to the operand address. The codes contained in these bits and their definitions are shown in Table 2-6.

Table 2-4. Standard Device Address Assignments

ADDRESS	DEVICE
00 01	NOT USED
02	REAL TIME CLOCK (RTC 61) 60Hz FIXED
03	REAL TIME CLOCK (RTC 61) 1000Hz FIXED
04	CARD PUNCH
05 06	ANALOG INPUT (ANI)
07	ANALOG OUTPUT (ANO)
08	CONTACT CLOSURE INPUT (CCI)
09	CONTACT CLOSURE OUTPUT (CCO)
0A 0B	SYNCHRONOUS COMMUNICATION DEVICE (SYNCH COMM)
0C	LINE PRINTER (LP)
0D	CARD READER (CR)
0E	PAPER TAPE PUNCH (PTP)
0F	PAPER TAPE READER (PTR)
10	TELETYPE (TTY)
11	UNASSIGNED
12 13	ASYNCHRONOUS COMMUNICATION DEVICE (ASYNC COMM)
14	CARD PUNCH #2
15 16	ANALOG INPUT #2
17	ANALOG OUTPUT #2
18	CONTACT CLOSURE INPUT #2
19	CONTACT CLOSURE OUTPUT # 2
1A 1B	SYNCHRONOUS COMMUNICATION DEVICE # 2
1C	LINE PRINTER #2
1D	CARD READER # 2
1E	PAPER TAPE PUNCH #2
1F	PAPER TAPE READER #2
20	TELETYPE # 2
21	UNASSIGNED
22 23	ASYNCHRONOUS COMMUNICATION DEVICE #2
24	CARD PUNCH # 3
25 26	ANALOG INPUT #3
27	ANALOG OUTPUT # 3
28	CONTACT CLOSURE INPUT # 3
29	CONTACT CLOSURE OUTPUT #3
2A	UNASSIGNED
2B	UNASSIGNED
2C	UNASSIGNED
2D	UNASSIGNED
2E	CIC
2F	INT #1

ADDRESS	DEVICE
30	TELETYPE # 3
31	UNASSIGNED
32 33	ASYNCHRONOUS COMMUNICATION DEVICE # 3
34	CARD PUNCH #4
35 36	ANALOG INPUT # 4
37	ANALOG OUTPUT #4
38	CONTACT CLOSURE INPUT #4
39	CONTACT CLOSURE OUTPUT # 4
3A 3B	DIGITAL INPUT/OUTPUT # 2
3C	UNASSIGNED
3D	UNASSIGNED
3E	CIC
3F	INT #2
40 ↓ 4F	FIXED HEAD DISC (FHD)
50 ↓ 5F	MOVING HEAD DISC (MHD)
60 ↓ 6F	MAGNETIC TAPE (MT)
70 71	COMMUNICATIONS CONTROL CONSOLE
72	AUTO-RESET
73 ↓ 77	NOT USED
78	LAST PARITY ERROR ADDRESS
79	MEMORY VIOLATION ADDRESS
7A	LOWER LIMIT ADDRESS (MEMORY PROTECT)
7B	UPPER LIMIT ADDRESS (MEMORY PROTECT)
7C	LAST BUFFERED I/O DEVICE TIMEOUT
7D	LAST DIRECT I/O DEVICE TIMEOUT
7E	BUFFER OVERFLOW (DEVICE ADDRESS)
7F	DEVICE INTERRUPT NUMBER

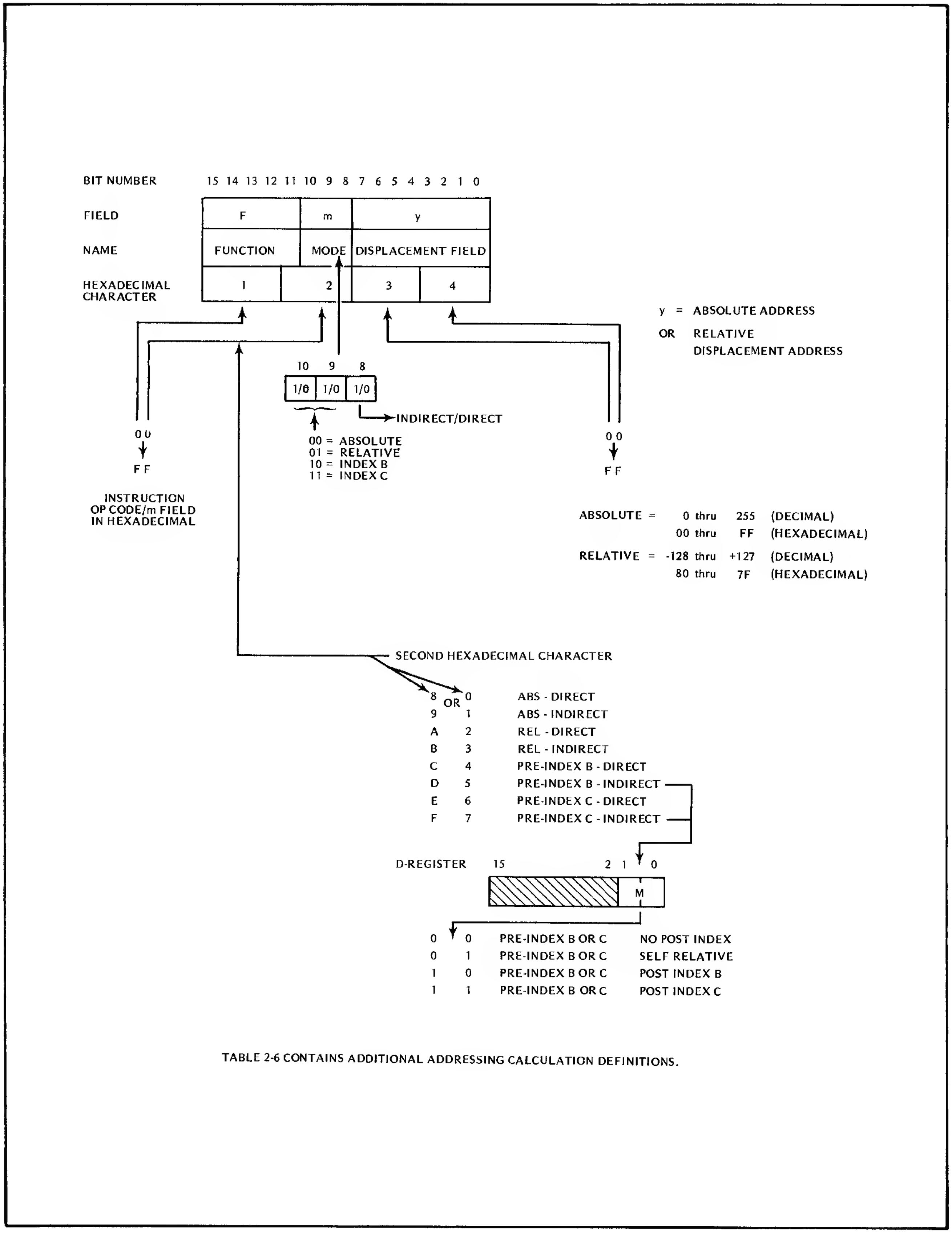


Figure 2-5. Word Format – Memory Reference Instruction

Table 2-5. Mode Field Codes

m FIELD BITS			ADDRESS CALCULATION
10	9	8	
0	0	0	DIRECT - ABSOLUTE
0	1	0	DIRECT - RELATIVE
1	0	0	DIRECT - PRE-INDEX B
1	1	0	DIRECT - PRE-INDEX C
0	0	1	INDIRECT - ABSOLUTE
0	1	1	INDIRECT - RELATIVE
1	0	1	INDIRECT - PRE-INDEX B
1	1	1	INDIRECT - PRE-INDEX C

Displacement (y) Field

Bits 0 through 7 of the instruction word are the displacement field. The decoding of y is determined by the setting of the m field. When m specifies an absolute address, y is an absolute address between 0 and 255 (X'00' through X'FF'). When m specifies a relative address, y has a value relative to the P-register from -128 through +127 (X'80' through X'7F').

Table 2-6. Designator Register Post Index Mode Bits
(Pre-Index, Indirect Only)

BITS		POST INDEX MODE
1	0	
0	0	NO POST MODIFICATION
0	1	OPERAND ADDRESS IS SELF RELATIVE
1	0	POST INDEXING REQUIRED - B INDEX
1	1	POST INDEXING REQUIRED - C INDEX

MEMORY ADDRESSING

The memory addressing scheme for the Westinghouse 2500 uses the three- bit code described under Word Format to specify address calculation. This code specifies addressing operations which can range from directly addressing the first 256 (000-255) locations in memory to indirectly addressing any location within the 64K memory capacity. A diagram of the addressing capabilities is shown in Figure 2-6. A summary of these addressing schemes is shown in Table 2-7. Each scheme is explained in detail in subsequent paragraphs.

Direct Addressing

If bit 8 in the mode field of a memory reference instruction is a ZERO, direct addressing is specified. In this mode, the calculated memory address is the location containing the operand of the instruction. The types of calculations used to obtain this direct address are:

- Absolute

- Relative
- Indexed

If bit 10 of the mode field is ZERO, then bit 9 specifies absolute (bit 9=0) or relative (bit 9=l) addressing. If bit 10 of the mode field is ONE, then indexing operations are specified and bit 9 indicates whether the B-index register (bit 9=0) or the C-index register (bit 9=1) is used for calculations.

Direct Absolute Addressing——In this mode of addressing, the 8-bit displacement field contains the address of the operand. The addressing capability of this field ranges from location X'00' through X'0FF' (256 decimal locations).

Direct Relative Addressing——In this mode of addressing, the 8-bit displacement field is used to calculate the operand address relative to the address of the instruction being executed (contents of the P-register). Thus, the displacement field must contain a sign bit (bit 7) and a relative displacement (bits 6- 0). If the sign bit is ONE, then the displacement may indicate -1 to -128 decimal locations expressed in two's-complement. If the sign bit is ZERO, the displacement may be +1 to +127 decimal locations. In either case, the displacement is sign-extended from bit 7 and algebraically added to the contents of the P-register to obtain the operand address.

Direct Indexed Addressing——In this mode of addressing, the 8-bit (absolute) displacement field is added to the contents of the 16-bit index register specified (B or C) and the result is used to obtain the operand from memory. Since the index register has a 16-bit capacity, this addressing method may be used for addressing any location within the 64K capacity of the machine.

Indirect Addressing

If bit 8 in the mode field of a memory reference instruction is a ONE, indirect addressing is specified. In this mode, memory must be accessed twice to obtain the operand. The first access of memory obtains the address of the operand and the second access obtains the operand. Obtaining the operand address in the indirect mode utilizes the same methods as direct mode does for obtaining the operand. That is, bits 9 and 10 are used in the same manner to specify absolute, relative or indexed calculations. These calculations and resulting memory addressing schemes are performed exactly the same as the direct addressing; however, if indexing operations are specified, post-indexing operations may be required and these operations are indicated by bits 1 and 0 of the Designator (D) register.

Post Index Operations——After the operand address is obtained through the pre-indexing operation, the address may be modified as specified by the Post- Index Mode bits (1 and 0) of the Designator Register. These mode bits are designated M in Table 2-6. If the M bits are both ZERO, the operand address is used, without alteration to address memory and fetch the operand. If bit 0 is ONE and bit 1 is ZERO, the operand address is calculated by adding the contents of the location fetched by the pre-indexing operation to the address location (Indirect Self-Relative). If bit 1 of the Designator register is ONE, then post-indexing calculations must be performed and bit 0 indicates the index register to be used (ZERO=B index, ONE=C index). The contents of the specified index register are algebraically added to the contents of the location fetched by the pre-index operation and the result is the operand address.

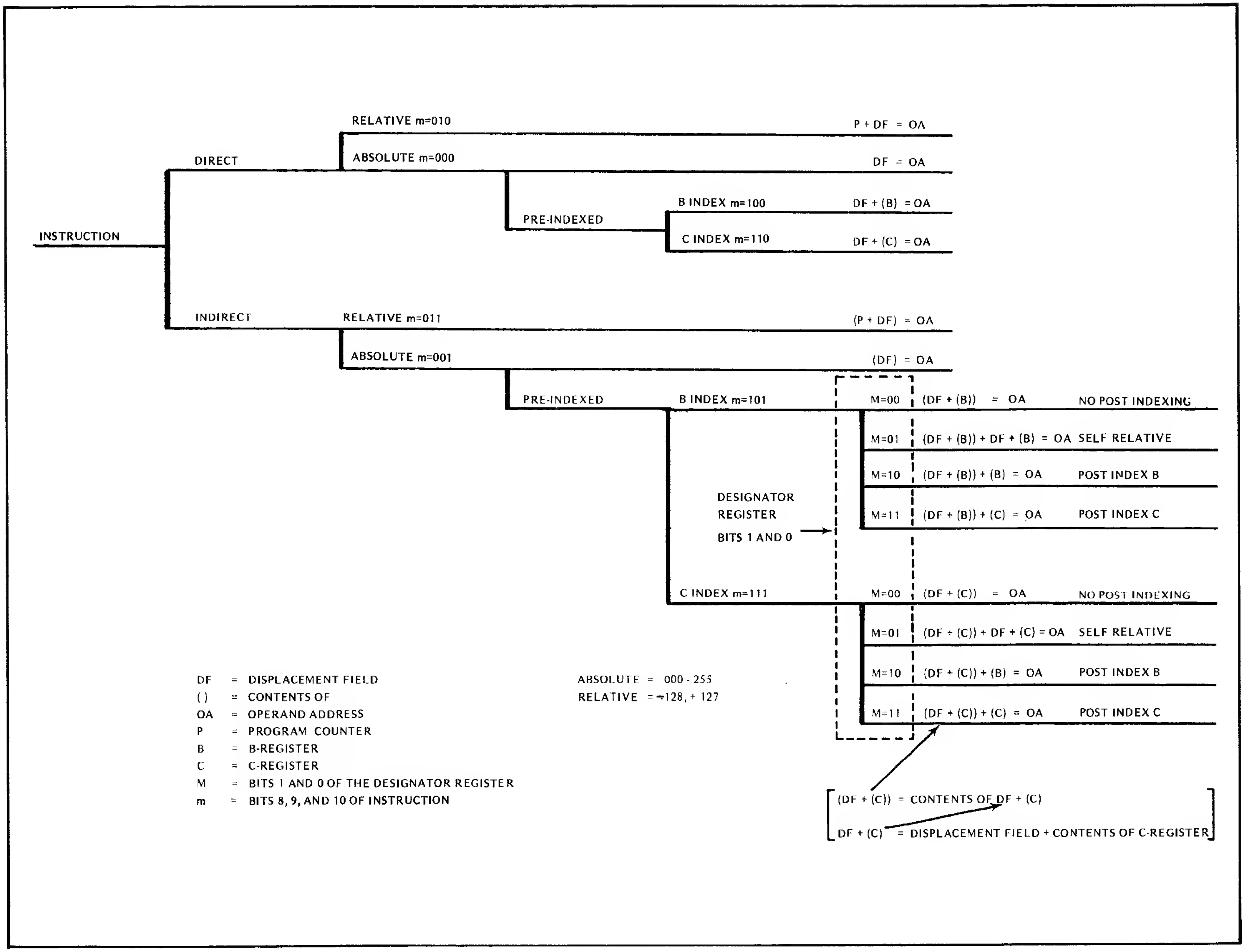


Figure 2-6. Addressing Schemes

Table 2-7. Address Calculation Schemes

ADDRESSING MODE	m			M		OPERAND ADDRESS CALCULATION
	10	9	8	1	0	
DIRECT – ABSOLUTE	0	0	0	–	–	DF = OA
DIRECT – RELATIVE	0	1	0	–	–	(P) + DF = OA
DIRECT – INDEXED B	1	0	0	–	–	DF + (B) = OA
DIRECT – INDEXED C	1	1	0	–	–	DF + (C) = OA
INDIRECT – ABSOLUTE	0	0	1	–	–	(DF) = OA
INDIRECT – RELATIVE	0	1	1	–	–	((P) + DF) = OA
INDIRECT – PRE-INDEXED B	1	0	1	0	0	(DF + (B)) = OA
INDIRECT – PRE-INDEXED B, SELF RELATIVE	1	0	1	0	1	(DF + (B)) + DF + (B) = OA
INDIRECT – PRE-INDEXED B, POST-INDEXED B	1	0	1	1	0	(DF + (B)) + (B) = OA
INDIRECT – PRE-INDEXED B, POST-INDEXED C	1	0	1	1	1	(DF + (B)) + (C) = OA
INDIRECT – PRE-INDEXED C	1	1	1	0	0	(DF + (C)) = OA
INDIRECT – PRE-INDEXED C, SELF RELATIVE	1	1	1.	0	1	(DF + (C)) + DF + (C) = OA
INDIRECT – PRE-INDEXED C, POST-INDEXED B	1	1	1	1	0	(DF + (C)) + (B) = OA
INDIRECT – PRE-INDEXED C, POST-INDEXED C	1	1	1	1	1	(DF + (C)) + (C) = OA

DF = DISPLACEMENT FIELD
( ) = CONTENTS OF
OA = OPERAND ADDRESS
P = PROGRAM COUNTER
B = B INDEX REGISTER
C = C INDEX REGISTER

Operand Addressing Summary

1. All displacement addresses specified by the machine instruction word in bits 7 through 0 are either absolute in the range 0-255(X'00'-X'FF') or are relative addresses within -128 (X'80') to +127 (X'7F') locations of the current instruction (P relative).

2. This address can be utilized as a direct address (the operand is obtained directly from the address location specified) or it can be

taken as an indirect address. In this case, the indirect address accessed by the displacement field contains a 16-bit address which is then the operand address.

3. Index registers B and C can add a bias to an absolute address in either the direct or indirect modes. This is called pre-indexing and it is added to the displacement field to obtain the operand address or address of the operand address.

4. When pre-indexing and indirect are specified, the setting of the D register bits 1 and 0 determine how the operand address is obtained.

INTERRUPTS

Westinghouse 2500 interrupts are structured in the following manner:

1. Processor Interrupts (PI) have the highest priority. There are two Processor Interrupts: one is generated by a Power Failure or a Memory Parity Error and the other is generated by:

   - Memory Write Violation
   - Instruction Violation
   - Auto Restart
   - Pressing START after pressing RESET without accessing the P register.

2. External Interrupts (EI) from the I/O subsystem have their priority determined by the physical location of their I/O interface card in the processor. If an External Interrupt and a Processor Interrupt occur simultaneously, the Processor Interrupt has priority.

3. Service Request Interrupts (SRI) have the lowest priority and are utilized by Buffered I/O devices to transfer data. Each peripheral which uses the buffered I/O transfer method has a pair of memory locations assigned to it. The first location usually contains an OTR or ITR instruction which will transfer a 16-bit data word each time a Service Request Interrupt (SRI) is generated. The second location contains a word count which keeps track of how many data words are transferred. At the completion of the transfer, the Buffer Overflow Interrupt is generated.

When an interrupt signal is detected by the CPU, the current instruction is completed, control is transferred to one of the dedicated memory locations shown in Table 2-8 and an out-of-sequence instruction is executed. Executions from any of these locations involve a modified addressing scheme. If the indirect bit (bit 8) of the instruction contained in one of these locations is set to ONE, the CPU will add X'100' to the displacement field before addressing memory to obtain the operand address. Direct addressing (bit 8=0) remains unchanged.

The Store Status (SST) and Enter Status (EST) instructions are provided so that interrupts can be serviced efficiently. Normally, except for SRI locations, a program should have an SST instruction stored at each memory location which is dedicated to an interrupt.

When an SST instruction is executed, the status of the processor is stored in seven memory locations and control is transferred to the address specified by the SST instruction displacement field. The EST instruction provides a means to restore the machine status that existed before the interrupt occurred and then to transfer control to the instruction that was to be next executed in the interrupted program.

Section 4 contains a detailed description of both the SST and EST instructions.

Table 2-8. Interrupt Priorities

TYPE OF INTERRUPT	PRIORITY	LOCATION
PROCESSOR INTERRUPT MEMORY WRITE VIOLATION (D12) INSTRUCTION VIOLATION (D13)	1	X'103'
PROCESSOR INTERRUPT POWER FAILURE (D15) MEMORY PARITY ERROR (D14) AUTO RESTART MANUAL START IMMEDIATELY FOLLOWING RESET WITHOUT DISPLAYING P-REGISTER	2	X'101'
EXTERNAL INTERRUPT CONSOLE INTERRUPT PERIPHERAL DEVICES	3	X'100'
SERVICE REQUEST INTERRUPT BUFFER OVERFLOW	4	X'102'
SERVICE REQUEST INTERRUPTS DEVICE SRI	5 THROUGH 63	X'104' THROUGH X'17F'

Processor Interrupts

There are two Processor Interrupts: the one of higher priority is generated by a Memory Write Violation or an Instruction Violation, and the other is generated by a Memory Parity Error or Power Failure.

Memory Write Violations or Instruction Violations set their respective designator bits in the D-register (Memory Write Violation, bit 12; Instruction Violation, bit 13) and cause the instruction in location X'103' to be executed out of sequence.

A Memory Parity Error or Power Failure will set their respective designator bits in the D-register (Memory Parity Error, bit 14; Power Failure, bit 15) and cause the instruction in location X'101' to be executed out of sequence.

During the out-of-sequence execution of either Processor Interrupt location, the Designator register is unaltered but the instruction is executed as if the CPU is in the privileged mode. At the end of the instruction execution, the Memory Write Lockout designator (D-register, bit 8) and the Privileged Lockout designator (D-register, bit 9) are reset to ZERO and the interrupt lockouts are set.

If the out-of-sequence instruction is an SST, the D-register image stored may be interrogated to determine the cause of the Processor Interrupt, and the stored P-register image will contain the address of the instruction executed just prior to the interrupt. Execution of the SST will set the interrupt lockouts, reset the Memory Write and Privileged Instruction lockouts, and reset only those Processor Interrupt designators associated with the interrupt.

The Processor Interrupt lockout must then be reset by a CDR instruction. Thus, both Processor Interrupts will be serviced in order of priority if they occur simultaneously, and if a Processor Interrupt occurs during the service routine, it will be serviced when the CDR is executed to reset the Processor Interrupt lockout.

Memory Write Violation——A Memory Write Violation occurs when any instruction is executed which attempts to modify a protected memory location (see Memory Protect Option). If the memory word address is not in the allowed range, the write is not permitted, Processor Interrupt is generated, and the violation address is stored in a 16-bit hardware register addressed as Direct I/O device X'79'. Bit 12 of the designator register is set.

Instruction Violation——Any of the following operations will cause an Instruction Violation and will set bit 13 of the designator register:

1. The execution of a CDR to initiate an extended op-code with the displacement field containing an un-implemented op-code.

2. The attempted execution of any privileged instruction when the CPU is operating in the non-privileged mode.

3. The attempted execution of an ITR or OTR instruction from any location other than an SRI.

In the first instance, the CDR will be executed and the word following the CDR will be fetched before the Processor Interrupt is generated. In the second and third instances, the instruction will not be executed and a Processor Interrupt will be generated.

Power Failure——When voltage falls below a certain point for a pre-determined period of time, the power failure designator (D-register, bit 15) is set and a Processor Interrupt is generated. Within two milliseconds after the Processor Interrupt is generated, the core memory power supply is shut down to prevent half-setting cores.

Memory Parity Error——Memory parity (odd) checking is performed during the memory read operation. Every memory cycle consists of a read/write or read/modify/write operation, so any instruction which access memory may result in a Memory Parity Error. If a Memory Parity Error is detected, the appropriate designator (D- register, bit 14) is set and a Processor Interrupt is generated.

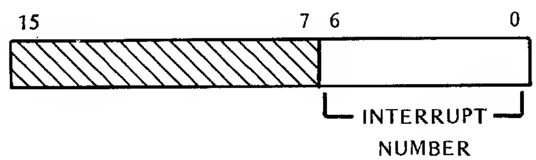
External Interrupt

When an External Interrupt (EI) is received from the I/O subsystem and the EI lock-out is reset (Designator register, bit 11=0), the instruction in X'100' is executed out-of-sequence. If an EI and a PI occur simultaneously, the PI has priority. During the out-of-sequence execution, privileged mode is assumed, even through the lock-outs are not reset until the end of the execution. At the completion of the out-of-sequence instruction, the lock-out for EI is set (designator register, bit 11=1) and the privileged mode is entered.

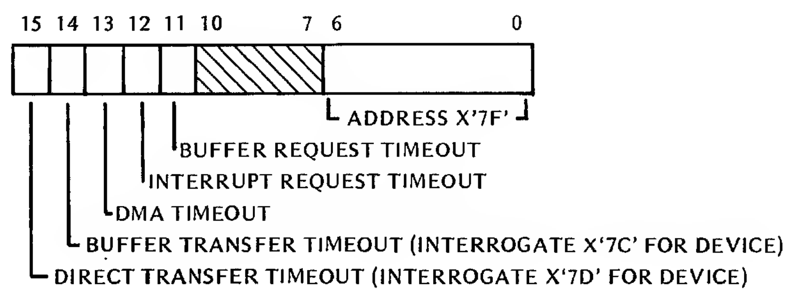
Device priorities in the external interrupt structure are assigned by their relative position in a daisy-chain. The physical location of the device controller card determines the priority in

the chain. The Console Interrupt is one of the External Interrupts and is usually assigned the first priority.

When an external interrupt is detected by the CPU, an out-of-sequence execution of the instruction stored at location X'100' is performed. Generally, this will be an SST instruction so that the CPU's status will be saved. At this point in time, all that is known is that an external interrupt has occured. It is now necessary for the external interrupt, sub-routine to interrogate device address X'7F' by means of an IOA instruction to determine which device has caused an interrupt. (Note, if X'7F' is not read, interrupts will continue to be generated.) Upon conclusion of the IOA instruction, the accumulator will contain a word in the following format:



If the interrupt number is X'7F', the word in the accumulator will be further defined in bits 11 through 15:



If the standard device address assignments are used in the system, then bits 0 through 6 contain an interrupt number that corresponds to the device standard address listed in Table 2-4.

In addition to the standard device addresses, number X'7E' is defined as the console interrupt number. Once the interrupting device is determined, the program branches to the sub-routine for that device.

If the cause of the interrupt is a buffer request timeout, an interrupt request timeout, or a DMA timeout, the interrupt subroutine will not be able to directly determine the causing device as the I/O subsystem does not have any provision to save the address of the device(s) that is timing out. The only thing that can be identified under these three conditions is the type of interrupt (bits 11, 12, and 13 of the X'7F' status word).

The direct transfer timeout is described in the section covering direct transfer and the buffer transfer timeout is described in the section covering buffer transfer.

Service Request Interrupt (SRI)

The SRI number assigned to a device is usually the same as its direct I/O number. For example, the Teletype is assigned direct I/O number X'010' and SRI number X'010'. Additionally, each device's Service Request Interrupt (SRI) is hardware-assigned two memory locations which are defined as X'100' + (2 x SRI No.) and X'101' + (2 x SRI No.):

Location No. 1

X'100' + (2 x SRI No.)	usually contains an OTR (output) or ITR (input) instruction.

Location No. 2

X'101' + (2 x SRI No.)	usually contains a number which is the two's complement of the buffer size (negative index count).

The operand address calculation for the first word is performed as follows:

1. X'100' + first instruction's displacement field = indirect address.

2. Contents of indirect address + second word's negative address count = operand address.

After the operand is calculated, the instruction is executed as an out-of-sequence instruction. It should be noted that normally the first memory location contains an ITR or OTR instruction but any instruction in the first word is executed as an out-of-sequence instruction and the running program resumed. The buffered I/O and SRI sequence is described in the section on Buffered I/O Transfers.

Interrupt Suppression

Interrupts are suppressed immediately after the execution of certain transfer instructions. This suppression occurs only between the transfer instruction and the following instruction; this allows a subroutine to lockout various interrupts on its first instruction. The suppression occurs under the following conditions:

SRI Suppression——SRIs are suppressed immediately after the following command executions:

1. SST: PLO=0

2. SRI out-of-sequence execution

3. EI out-of-sequence execution

EI Suppression——EIs are suppressed immediately after the following command executions:

1. All jumps: CJP, JMP, NJP, OJP, PJP, ZJP; if PLO=0

2. SST: PLO=0

3. STP: PLO=0

4. Single Instruction mode

PI Suppression——PIs are not suppressed immediately after the execution of any command.

PROTECTION FEATURES

Un-Implemented Instruction Trap

If an un-implemented extended function code instruction is encountered in a program, the CPU generates a Processor Interrupt and sets the Instruction Violation designator (bit 13) in the D-register.

Power Failure Interrupt

If there is a power outage, a high-priority Processor Interrupt is generated by the power failure interrupt circuitry. With standard software, this interrupt initiates a routine which saves register contents, designator settings, and (at the programmer's option) arms the auto-restart circuit.

OPTIONAL FEATURES

Memory Parity Error Detection

Memory parity (odd) is generated by the memory subsystem. If a parity error is detected by the subsystem, the CPU generates the Processor Interrupt and sets the Memory Parity Error designator (bit 14) in the D-register.

Memory Protect

The Memory Protect option consists of: Memory Write Protection and Privileged Instruction Mode. Use of this option allows a monitor system to operate in the Privileged Instruction Mode and limits user programs (or tasks) to writing in a monitor-defined area of core in the Non-Privileged Mode.

Memory Write Protect——The memory write protect option consists of three hardware registers (direct I/O addresses 79, 7A and 7B). Two of these registers are preset by IOA instructions (in Privileged Mode) to establish the unprotected lower limit (7A)and upper limit (7B). These limits define, inclusively, the unprotected area of memory. Once the limits are set, protection of memory outside these limits is initiated by setting either bit 8 (Write Lockout) or bit 9 (Privileged Lockout) of the Designator register. If an out of bounds write is attempted, after write protect is initiated, the write is not permitted, a Processor Interrupt is generated, and the violation address is stored in the third hardware register (direct I/O address 79).

Non-Privileged Instruction Mode——When initial power is applied to the Westinghouse 2500, the machine is placed in the privileged mode of operation. Under privileged mode conditions, all instructions are executed without restriction. When the computer is placed in the non-privileged mode, certain instructions cannot be executed. These instructions include:

1. All I/O instructions.
2. The STP (stop) instruction.
3. Three functions of the CDR instruction (see Section 4).

The non-privileged mode of operation is in effect whenever the Privileged Instruction Lockout designator (bit 9) of the D-register is set. There are several ways of setting this bit,

however, the most desirable method is by executing an EST (Enter Status) instruction while in the privileged mode. If execution of one of the above mentioned instructions is attempted while in the non-privileged mode, the following events occur:

1. A no-op is executed.
2. The Processor Interrupt is generated.
3. When the interrupt is acknowledged, an out-of-sequence instruction (location X'0103') is executed.
4. This out-of-sequence instruction sets all interrupt lock-outs including the Processor Interrupt lockout.
5. The machine is placed in the privileged mode.

Protect Operation—Normally, each program executed in the Westinghouse 2500 is run as a task under a Monitor program. If protect is installed, only the Monitor is privileged and each task is assigned the core limits within which it may write. If a task attempts to write outside its assigned core area, a memory write violation is generated. If a task attempts to execute a privileged instruction, an instruction violation is generated.

If protect is not installed, all programs including Monitor are privileged and each program runs under the following conditions:

1. An attempt to execute a CDR to set the Privileged Instruction Lockout results in a no-op.
2. Accessing non-existent core generates a memory parity violation.
3. No instruction violations are generated except un-implemented extended op-codes.

Auto Restart

After a power interruption, if the auto-restart has been armed, a start signal is generated. The parameters that were saved by the power failure interrupt then can be used to restore the program that was running before the power failure occurred.

Floating Point Arithmetic Unit

The Westinghouse 2500 Floating Point Arithmetic option provides hardware single precision addition, subtraction, multiplication and division. The data format is the same as described in the Westinghouse 2500 FORTRAN IV Reference Manual. Each floating point operation is initiated by a function code which is transferred to the floating point unit from the CPU by the execution of a CDR instruction.

Real Time Clock Options

The real time clock option issues Service Request Interrupts to the CPU at 60 and 1000 Hz frequencies. The 60 Hz clock is addressed as peripheral device X'02' and the 1000 Hz clock is addressed as peripheral device X'03'.

SECTION 3

OPERATOR'S CONTROL PANEL

PANEL CONTROL AND INDICATORS

The Westinghouse 2500 Control Panel is shown in Figure 3-1. The controls and indicators on this panel are listed and described in Table 3-1. Using the control panel, an operator may enter the optional hardware bootstrap, initiate loading of the absolute binary loaders, start a program, stop a program, step through a program, display and alter the general program registers.

ENTER THE HARDWARE BOOTSTRAP

Machines containing the hardware bootstrap are initialized by the following procedure:

1. Turn the keyswitch to ON.
2. If a device other than the standard ASR is to be used to input, the device address will have to be changed. Eight toggle switches are attached to the hardware bootstrap to permit this change.
3. Raise, then press the RESET/BOOTSTRAP switch.
4. Refer to the Loader Reference Manual (Publication No. 25REF-003) for instructions on loading the Absolute Binary Loader (Loader III).

MANUALLY ENTERED BOOTSTRAP

If the hardware bootstrap option is not used, the bootstrap program can be entered manually. The manually entered bootstrap consists of ten hand loaded instructions and are listed after the following procedure. To load the program perform the following sequence of operations:

1. Turn the keyswitch to ON.
2. Set the SWITCH REGISTER to the first address (X'0000').
3. Press the LOAD ADDR switch.
4. Set the SWITCH REGISTER to the desired data word value (see Table 3-2 or 3-3). For example the first data word value will be X'0005'.
5. Raise the LOAD S switch.
6. Set the SWITCH REGISTER to the next address to be loaded.

7. Repeat steps 3 through 6 until the entire program is entered.

8. Refer to the Loader Reference Manual (Publication No. 25REF-003) for instructions on loading the Absolute Binary Loader (Loader III).

Figure 3-1. Westinghouse 2500 Control Panel

DISPLAY MEMORY

Locations in memory not provided with display switches on the control panel may be displayed using the following procedures:

1. Turn the keyswitch to the ON position.

2. Raise the RESET switch.

3. Set the desired memory location address in the SWITCH REGISTER.

4. Press the LOAD ADDR switch. The address lights indicate the desired address and the display lights show the contents of the address specified.

LOAD/ALTER MEMORY

Locations in memory not provided with load switches on the control panel may be altered or loaded using the following procedures:

1. Turn the keyswitch to the ON position.

2. Raise the RESET switch.

3. Set the SWITCH REGISTER to the address of the desired memory location.

4. Press the LOAD ADDR switch. The address lights should indicate the desired address.

5. Set the SWITCH REGISTER with the desired contents to be loaded into the indicated memory location.

6. Raise the LOAD [S] switch.

7. Press the DISPLAY [S] switch. The display lights should now show the new contents of the specified memory location.

LOAD A PROGRAM

To manually load a program, perform the following steps:

1. Turn the keyswitch to the ON position.

2. Raise the RESET switch.

3. Set the SWITCH REGISTER to the desired starting location.

4. Press the LOAD ADDR switch.

5. Set the SWITCH REGISTER with the data word to be entered.

6. Raise the LOAD [S] switch.

7. Raise the INC. & DIS switch.

8. Repeat Steps 5, 6 and 7 until the program is entered.

START A PROGRAM

To start a program perform the following steps:

1. Turn the keyswitch to the ON position.

2. Raise the RESET switch.

3. Set the SWITCH REGISTER to the program starting address, minus one.

4. Raise the LOAD P switch.

5. Set the INSTRUCTION switch for the desired mode of operation (if this switch is raised, the START switch will execute one instruction at a time).

6. Press the START switch.

Table 3-1. Panel Controls and Indicators

DESIGNATOR	TYPE	FUNCTION
OFF/PWR ON/LOCK	Key Switch	Turns power on and off, locks out all control switches except CONS INT and lamp test.
ADDRESS	16 Indicators	Displays the 16-bit content of the S register.
DISPLAY	16 Indicators	Displays the 16-bit content of the addressed location.
SWITCH REGISTER	16 Toggle Switches	Sets data word for input to memory or the hardware registers.
INCR & DISP/LOAD ADDR	Double Throw Momentary	Momentarily raising INCR & DISP switch increments the S register by one and displays the contents of the newly addressed location. Momentarily moving the switch to the LOAD ADDR position loads the S register with the contents of the SWITCH REGISTER and displays the contents of the addressed location.
LOAD/DISPLAY [S]	Double Throw Momentary	Momentarily moving the switch to the LOAD position enters the contents of the SWITCH REGISTER into the memory location specified by the address display. Momentarily moving the switch to the DISPLAY position displays the location specified by the address display.
LOAD/DISPLAY P,B,C,E,A	Double Throw Momentary	Momentarily moving the switch to the LOAD position enters the contents of the switch register into the specified register. Momentarily moving the switch to the DISPLAY position displays the contents of the specified register.
LOAD/DISPLAY D	Double Throw Momentary	Momentarily moving the switch to the LOAD position enters the status of the switch register into the designator (D) register. The bit designations are: 0 Post Index Mode 1 Post Index Mode 2 Zero Designator 3 Positive Designator 4 Overflow Designator 5 Carry Designator 6 not assigned (not loadable) 7 not assigned (not loadable) 8 Memory Write Lockout Bit 9 Privileged Instr. Lockout Bit 10 SR Lockout Bit 11 External Interrupt Lockout Bit 12-15 Processor Interrupt Section (not loadable) 12 Memory Violation Bit 13 Instruction Violation Bit 14 Parity Bit (Memory) 15 Power Failure Bit

Table 3-1. Panel Controls and Indicators (Continued)

DESIGNATOR	TYPE	FUNCTION
LOAD/DISPLAY D (Continued)	Double Throw Momentary	Momentarily moving the switch to the DISPLAY position displays the contents of the designator (D) register.
DISPLAY F	Single Throw Momentary	The function (F) register cannot be loaded through the operator's panel. Momentarily moving the switch to the DISPLAY position displays bits 8 through 15 (Function and Addressing Mode) of the Function (F) register.
RUN	Indicator	When lighted, indicates the CPU is running a program. When unlighted indicates that the CPU is stopped.
HALT/START	Double Throw Momentary	Momentarily depressing this switch will start the CPU at P + 1. The CPU will run the program or perform a single instruction depending on the setting of the INSTRUCTION switch. Momentarily raising this switch to the HALT position, stops the CPU after the current instruction (no other changes are made).
RESET/BOOTSTRAP	Double Throw Momentary	Momentarily raising this switch to the RESET position halts the CPU instantanously and performs the following: 1. The designator (D) register is cleared, the external interrupt lockout, and SRI lockout designators are set. 2. All interrupts are cleared. 3. Resets floating point hardware. 4. The I/O subsystem is reset. 5. The sequencer is placed in the idle state (0,0). After raising RESET, the operator has four options: 1. Press START. This forces a Processor Interrupt to location X'101' if P is not accessed before pressing start. 2. Press DISPLAY P to condition the CPU, and then press START (no PI occurs). P + 1 is executed. 3. Alter the contents of P (this also conditions the CPU) and then press START. P + 1 is executed. 4. Press BOOTSTRAP and load the bootstrap program into the dedicated memory locations. BOOTSTRAP should be pressed only after RESET is raised.
SNGL INSTR/LAMP TEST	Double Throw Toggle	Raising this switch allows stepping through a program, by performing one instruction each time the START switch is depressed. It should be noted the external interrupts are suppressed when single instruction is active. The CPU

Table 3-1. Panel Controls and Indicators (Continued)

DESIGNATOR	TYPE	FUNCTION
SNGL INSTR/LAMP TEST (Continued)	Double Throw Toggle	executes the program without stopping when the switch is in the center position. Depressing this switch turns on all lamps on the operating panel.
CONS INT	Single Throw Momentary	Depressing this switch will generate an External Interrupt through the I/O subsystem.

Table 3-2. Manually Entered Paper Tape Bootstrap

LOCATION	CONTENTS	COMMENTS
X'0000'	X'0005'	
X'0005'	X'5000'	
X'0006'	X'31E0'	
X'0007'	X'8B02'	
X'0008'	X'6800'	
X'0009'	X'5BYY'	YY = 10 for ASR, 0F for PTR
X'0029'	X'00YY'	
X'0034'	LWA+1	= 1000 for Loader III in page of 4K.
X'0102'	X'7181'	
X'0100' + 2YY	X'9080'	Loc = X'0120' for ASR, X'011E' for PTR.
X'0101' + 2YY	one's complement of word count	= X'FF00' for Loader III
X'0180'	LWA+1	= 1000 for Loader III in last page of 4K.
X'0181'	Transfer Address	= X'0EFF' for Loader III in 4K.

Table 3-3. Manually Entered Card Reader Bootstrap

LOCATION	CONTENTS	COMMENTS
X'0000'	X'0005'	
X'0006'	X'88FF'	
X'0007'	X'3IA0'	
X'0008'	X'8B02'	
X'0009'	X'6800'	
X'000A'	X'5B0D'	
X'0029'	X'000D'	
X'0034'	LWA+1	= 1000 for Loader III in last page of 4K.
X'0100'	X'7005'	
X'0102'	X'7181'	
X'011A'	X'9080'	
X'011B'	one's complement of word count	= X'FEA0' for Loader III.
X'0180'	LWA+1	= 1000 for Loader III in last page of 4K.
X'0181'	Transfer Address	= X'0E9F' for Loader III in 4K.

SECTION 4

COMPUTER INSTRUCTIONS

INTRODUCTION

This section of the manual provides necessary information for calculating instruction execution times, and presents a detailed description and word format for each instruction.

INSTRUCTION EXECUTION TIMES

Instruction execution time consists of the instruction fetch time, operand address calculation time, and action execution time. The times listed by the instruction descriptions reflect the fetch and action execution times. Operand address calculation times are shown in Table 4-1.

Table 4-1. Address Calculation Times

ADDRESS MODE	TIME IN MICROSECONDS
DIRECT – ABSOLUTE	NONE
DIRECT – RELATIVE	NONE
DIRECT – INDEXED	.650
INDIRECT – ABSOLUTE	.900
INDIRECT – RELATIVE	.900
INDIRECT – PRE-INDEXED	1.550
INDIRECT – PRE-INDEXED, SELF RELATIVE	2.450
INDIRECT – PRE-INDEXED, POST-INDEXED	1.550

INSTRUCTION DESCRIPTIONS

The instruction descriptions are arranged in alphabetical order by mnemonics on the remaining pages of this section.

ADA   ADA

ADD DOUBLEWORD TO ACCUMULATOR

OP CODE:

C0 - C7

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

ADA algebraically adds a double-length word to the contents of the accumulator and extended accumulator and places the result into the accumulator and extended accumulator.

	Most Significant	Least Significant
	E-Reg	A-Reg
Plus	Operand Address	Operand Address + 1
Result	E-Reg	A-Reg

End-around carry is not provided and negative quantities are expressed in two's complement.

Bit 15 of the most significant half of the doubleword indicates sign: all other bits indicate magnitude.

The designators are affected by the result of the addition.

DESIGNATORS:

Zero	(D2)	-	Set if doubleword is all zeros
Positive	(D3)	-	Set if bit 15 of the E-Reg = 0
Overflow	(D4)	-	Set if E-Reg overflows
Carry	(D5)	-	Set if carry is required

TIMING:

2.0 microseconds

ADD   ADD

ADD WORD TO ACCUMULATOR

OP CODE:

40 - 47

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

ADD performs an addition of the contents of the accumulator and the contents of the calculated operand address. The result is stored in the accumulator. End-around carry is not provided and negative quantities are expressed in two's complement. The contents of the operand address are not changed.

SUMMARY EXPRESSION:

$(A) + (OA) \rightarrow A$

DESIGNATORS:

Zero	(D2)	-	Set if result is all zeros
Positive	(D3)	-	Set if result has bit 15 = 0
Overflow	(D4)	-	Set by a sign change of the accumulator.
Carry	(D5)	-	Set if carry is required

TIMING:

1.3 microseconds

AND AND

AND WITH ACCUMULATOR

OP CODE:

58 - 5F

FORMAT:

15	14	13	12	11	10 9 8	7 6 5 4 3 2 1 0
0	1	0	1	1	m	y, y*
Function					Mode	Displacement

DEFINITION:

AND performs a logical "AND" between the accumulator and the operand on a strict bit basis, that is, each bit in the accumulator is "ANDed" with the corresponding bit of the operand.

0 & 0 = 0
0 & 1 = 0
1 & 0 = 0
1 & 1 = 1

SUMMARY EXPRESSION:

(A) & (OA) → A

DESIGNATORS:

Zero (D2) - Set or cleared according to accumulator
Positive (D3) - Set or cleared according to accumulator
Overflow (D4) - Do not change
Carry (D5) - Do not change

TIMING:

1.3 microseconds

CDR CDR

CHANGE DESIGNATOR REGISTER (PRIVILEGED)

OP CODE:

30 - 37 (Special Meaning)

FORMAT:

15	14	13	12	11	10 9 8	7 6 5 4 3 2 1 0
0	0	1	1	0	m	See Definition
Function					Mode	Displacement

DEFINITION:

The CDR instruction is a multi-function instruction. That is, the mode field is used to indicate the desired function to be executed. Functions available are shown in the following table:

OP CODE	MODE	FUNCTION
30	0 0 0	Change post-index designators or clear overflow
31*	0 0 1	Change lockout designation*
32*	0 1 0	Reset processor interrupt lockout*
33*	0 1 1	Change lockout designation and reset processor interrupt lockout*
34	1 0 0	Initiate Extended Function Code (calculate operand from next word)
36	1 1 0	Initiate Extended Function Code (all 16 bits of next word is operand address)
* These are privileged instructions and are not executable when bit 9 of the D-register is set.		

The functions indicated by the mode field are defined as follows:

Change Post-Index Designators

When the mode field of the CDR instruction contains 000 and bit 2 of the displacement field is set, the status of bits 1 and 0 of the displacement field are copied into bits 1 and 0 of the Designator (D) register.

Clear Overflow Designator

When the mode field of the CDR instruction contains 000 and bit 3 of the displacement field is set, the overflow designator (bit 4 of the D-register) is reset to zero.

Change Lockout Designators (Privileged Instruction)

When the mode field of the CDR instruction contains a ONE in bit 8 and a ZERO in bit 10, the lockout designators will be

CHANGE DESIGNATOR REGISTER (PRIVILEGED) (Cont'd)

copied from the displacement field bits 0, 2, 4 and 6 as stipulated by displacement field bits 1, 3, 5 and 7 in the following manner:

y_1=1, copy y_0 into D_8 (Memory Write Lockout)
y_3=1, copy y_2 into D_9 (Privileged Instruction Lockout)
y_5=1, copy y_4 into D_{10} (Service Request Lockout)
y_7=1, copy y_6 into D_{11} (External Interrupt Lockout)

If bit 9 of the D-register is set (Non-Privileged Mode) prior to this instruction, the instruction will not be executed and an Instruction Violation will be generated to produce a Processor Interrupt.

Reset Processor Interrupt Lockout (Privileged Instruction)

When the mode field of the CDR instruction contains a ONE in bit 9 and a ZERO in bit 10, the Processor Interrupt Lockout will be reset to ZERO.

NOTE

The Processor Interrupt Lockout is set by the CPU hardware at the execution end of the instruction pointed to by the Processor Interrupt.

This instruction will not be executed (and will produce a Processor Interrupt) if bit 9 of the D-register (Privileged Instruction Lockout) is set.

Initiate Extended Function Code

When the mode field of the CDR instruction contains a ONE in bit 10, the displacement field is used to supply an extended function code. If bit 9 of the CDR instruction is ZERO, the word following the CDR is used to calculate the operand address in the normal manner where the function field (bits 15-11) is ZERO, bits 10-8 are the mode field, and bits 7-0 are the displacement field. If bit 9 of the CDR instruction is one, the operand address is taken to be the full 16-bit word following the CDR instruction. The extended function codes are defined below:

MNEMONIC	HEX CODE	DESCRIPTION
DSY	FE	Displays contents of memory location specified by operand address. The Display Register will remain unchanged until an operator panel operation is performed or another extended op code of X'FE' is executed. The Display Register is cleared when START is depressed.
RSR	FF	Loads operator panel switch register into location specified by operand address.

TIMING:

1.0 microseconds

CARRY JUMP

OP CODE:

F8 - FF

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

CJP interrogates the status of bit 5 (Carry Designator) in the D-register. If bit 5 is set, the calculated operand address is transferred to the P-register. If bit 5 is not set, the jump is not executed.

SUMMARY EXPRESSION:

OA $\rightarrow$ P, if D5 = 1

NEXT INSTRUCTION EXECUTED IF CARRY BIT IS SET:

Calculated Operand Address + 1

DESIGNATORS:

Zero	(D2)	-	No change
Positive	(D3)	-	No change
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

0.9 microseconds if no jump;
1.025 microseconds for jump taken

DECREMENT LOCATION

OP CODE:

68 - 6F

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

DCR accesses a 16-bit word located at the calculated operand address, decrements the word by one, and restores the word to the same location. The result of the decrementing operation is used to set the zero, positive carry and overflow designators. The accumulator is not affected by this instruction.

SUMMARY EXPRESSION:

(OA) - 1 → OA

DESIGNATORS:

Zero	(D2)	-	Set if word contains all zeros
Positive	(D3)	-	Set if bit 15 of result = 0
Overflow	(D4)	-	Set if overflow occurs
Carry	(D5)	-	Set if carry occurs

TIMING:

1.4 microseconds

DIVIDE

OP CODE:

D8 - DF

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

DIV algebraically divides the contents of the double-length accumulator by the contents of the calculated operand address. The quotient is placed into the A accumulator and the remainder is placed into the E register (the sign of the remainder is always the same as the sign of the dividend). Negative quantities are expressed in two's complement.

SUMMARY EXPRESSION:

(E,A)/(OA) Q = A
R = E

DESIGNATORS:

Zero	(D2)	-	Set if quotient is all zeros
Positive	(D3)	-	Set if bit 15 of A = 0
Overflow	(D4)	-	Set if quotient overflows accumulator
Carry	(D5)	-	Set equal to the sign of the remainder (1=negative, 0=positive)

TIMING:

26.0 microseconds

EOR EOR

EXCLUSIVE OR WITH ACCUMULATOR

OP CODE:

50 - 57

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
0 1 0 1 0	m	y, y*
Function	Mode	Displacement

DEFINITION:

Each bit position of the operand is exclusively "ORed" with each corresponding bit position of the accumulator. The result is left in the accumulator.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0

SUMMARY EXPRESSION:

$(A) + (OA) \rightarrow A$

DESIGNATORS:

Zero	(D2)	-	Set or cleared according to accumulator
Positive	(D3)	-	Set or cleared according to accumulator
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

1.3 microseconds

EST EST

ENTER STATUS (Privileged)

OP CODE:

E0 - E7

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
1 1 1 0 0	m	y, y*
Function	Mode	Displacement

DEFINITION:

EST loads the six program registers (P, B, C, G, E and A) and the D-register (bits 0 - 11, only) in that order, starting with the location specified by the calculated operand address and incrementing through the next six locations. If the CPU is operating in the privileged mode at execution time (D9=1), the EST instruction will not load bits 8, 9, 10 and 11 into the D-register.

SUMMARY EXPRESSION:

$(OA) \rightarrow P$
$(OA + 1) \rightarrow B$
$(OA + 2) \rightarrow C$
$(OA + 3) \rightarrow G$
$(OA + 4) \rightarrow E$
$(OA + 5) \rightarrow A$
$(OA + 6) \rightarrow D$

NEXT INSTRUCTION EXECUTED:

(OA) + 1

TIMING:

7.65 microseconds

INC INC

INCREMENT LOCATION

OP CODE:

60 - 67

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

INC accesses a 16-bit word located at the calculated operand address, increments the word by one, and restores the word to the same location. The accumulator is not affected by this instruction. The result of the incrementing operation is used to set the zero, positive, carry and overflow designators.

SUMMARY EXPRESSION:

$(OA) + 1 \rightarrow OA$

DESIGNATORS:

Zero	(D2)	-	Set if word contains all zeros
Positive	(D3)	-	Set if bit 15 of the result = 0
Overflow	(D4)	-	Set if overflow occurs
Carry	(D5)	-	Set if carry occurs

TIMING:

1.4 microseconds

IOA IOA

INPUT TO/OUTPUT FROM ACCUMULATOR (PRIVILEGED)

OP CODE:

88 - 8F (Special Meaning)

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	1	m			x	Device Address						
Function					Mode			I/O = Input/Output	Displacement						

DEFINITION:

IOA performs a 16-bit data transfer between the accumulator and the I/O subsystem. If direct addressing is used, bits 0 through 6 of the displacement field are used by the I/O subsystem to address channels 0 through 127.

If indirect addressing is used, bits 0 through 7 are used to calculate the operand address in the usual manner. Bits 0 through 6 of the calculated address will be used to address the device, bit 7 will indicate direction, and bits 8 through 15 may be used as a device function code. This code is specified in the reference manual for the specific device.

DESIGNATORS:

Zero	(D2)	-	Set or cleared depending on accumulator
Positive	(D3)	-	Set or cleared depending on accumulator
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

2.0 microseconds

ITR ITR

INPUT TRANSFER REQUEST (PRIVILEGED)

OP CODE:

90

FORMAT:

15	14	13	12	11	10 9 8	7 6 5 4 3 2 1 0
1	0	0	1	0		Relative Buffer End Address
Function					Mode	Displacement

DEFINITION:

ITR is a privileged instruction and is only executed as an out-of-sequence instruction in response to a Buffered I/O request from the I/O subsystem. If encountered in a normal program sequence, a no-op is performed and an Instruction Violation (PI) is generated. Normal address calculations are not performed. The displacement field is added to X'0100' and the result is used to obtain the Buffer End Address.

The next contiguous location after the ITR instruction must contain the buffer word count in two's complement form. This word is incremented by one and added to the Buffer End Address. The result is used to address memory for the data transfer.

DESIGNATORS:

Zero	(D2)	-	No change
Positive	(D3)	-	No change
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

4.325 microseconds

JMP JMP

JUMP UNCONDITIONAL

OP CODE:

70 - 77

FORMAT:

15	14	13	12	11	10 9 8	7 6 5 4 3 2 1 0
0	1	1	1	0	m	y, y*
Function					Mode	Displacement

DEFINITION:

JMP transfers the calculated operand address to the P-register, thus transferring program control to the instruction contained in the operand address plus 1.

SUMMARY EXPRESSION:

OA → P

NEXT INSTRUCTION EXECUTED:

OA + 1

DESIGNATORS:

Zero	(D2)	-	No change
Positive	(D3)	-	No change
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

1.025 microseconds

LOAD ACCUMULATOR

OP CODE:

28 - 2F

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
0 0 1 0 1	m	y, y*
Function	Mode	Displacement

DEFINITION:

LDA loads a 16-bit data word into the accumulator from the memory location specified by the calculated operand address.

SUMMARY EXPRESSION:

$(OA) \rightarrow A$

DESIGNATORS:

Zero (D2) - Set if word contains all zeros
Positive (D3) - Set if bit 15 of accumulator = 0
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

1.4 microseconds

LOAD B INDEX REGISTER

OP CODE:

08 - 0F

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
0 0 0 0 1	m	y, y*
Function	Mode	Displacement

DEFINITION:

LDB loads a 16-bit data word into the B index register from the memory location specified by the calculated operand address.

SUMMARY EXPRESSION:

$(OA) \rightarrow B$

DESIGNATORS:

Zero (D2) - Set if word contains all zeros
Positive (D3) - Set if bit 15 = 0
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

1.4 microseconds

LDC LDC

LOAD C INDEX REGISTER

OP CODE:

10 - 17

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

LDC enters a 16-bit data word into the C index register from the memory location specified by the operand address.

SUMMARY EXPRESSION:

(OA) → C

DESIGNATORS:

Zero (D2) - Set if word contains all zeros
Positive (D3) - Set if bit 15 = 0
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

1.4 microseconds

LDE LDE

LOAD EXTENDED ACCUMULATOR

OP CODE:

20 - 27

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

LDE enters a 16-bit data word into the extended accumulator from the memory location specified by the calculated operand address.

SUMMARY EXPRESSION:

(OA) → E

DESIGNATORS:

Zero (D2) - Set if word contains all zeros
Positive (D3) - Set if bit 15 = 0
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

1.4 microseconds

LDG LDG

LOAD SHIFT DESCRIPTION REGISTER (G)

OP CODE:

18 - 1F

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
0 0 0 1 1	m	y, y*
Function	Mode	Displacement

DEFINITION:

LDG enters a 16-bit data word into the G-register from the memory location specified by the calculated operand address. The format for this word is:

15 14 13 12	11 10 9 8 7 6 5	4 3 2 1 0
X X X 0		X X X X X
Shift Type		Shift Count

Bit 15	14	13	12	Hexadecimal	
0	0	0	0	0	Single Left Arithmetic
0	0	1	0	2	Single Left Circular
0	1	0	0	4	Single Right Arithmetic
0	1	1	0	6	Single Right Circular
1	0	0	0	8	Double Left Arithmetic
1	0	1	0	A	Double Left Circular
1	1	0	0	C	Double Right Arithmetic
1	1	1	0	E	Double Right Circular

The G-register is unaffected by the shift operation.

SUMMARY EXPRESSION:

(OA) → G

DESIGNATORS:

Zero (D2) - Set if word contains all zeros
Positive (D3) - Set if bit 15 = 0
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

1.4 microseconds

MPY MPY

MULTIPLY ACCUMULATOR

OP CODE:

D0 - D7

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
1 1 0 1 0	m	y, y*
Function	Mode	Displacement

DEFINITION:

MPY algebraically multiplies the contents of the A-register by the contents of the calculated operand address. The product is placed in the E-register and the A-register with the low-order bits in the A-register and the high-order bits with sign extension in the E-register. Negative quantities are expressed in two's complement. The zero and positive designators are set or cleared as indicated by the double-length product.

SUMMARY EXPRESSION:

(A) X (OA) → EA

DESIGNATORS:

Zero (D2) - Set if product is all zeros
Positive (D3) - Set if E-register bit 15 = 0
Overflow (D4) - No change
Carry (D5) - Cleared to zero

TIMING:

18.0 microseconds

NEGATIVE JUMP

OP CODE:

B8 - BF

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

NJP interrogates the status of bit 3 (Positive Designator) in the D-register. If bit 3 is set, the jump is not executed. If bit 3 is not set, the calculated operand address is transferred to the P-register.

SUMMARY EXPRESSION:

OA → P, if D3 = 0

NEXT INSTRUCTION
EXECUTED IF POSITIVE
NOT SET:

Calculated Operand Address + 1

DESIGNATORS:

Zero (D2) - No change
Positive (D3) - No change
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

0.9 microseconds
1.025 microseconds for jump taken

OVERFLOW JUMP

OP CODE:

78 - 7F

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

OJP interrogates the status of bit 4 (Overflow Designator) in the D-register. If bit 4 is set, the calculated operand address is transferred to the P-register. If bit 4 is not set, the jump is not executed.

SUMMARY EXPRESSION:

OA → P, if D4 = 1

NEXT INSTRUCTION
EXECUTED IF OVERFLOW
SET

Calculated Operand Address + 1

DESIGNATORS:

Zero (D2) - No change
Positive (D3) - No change
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

0.9 microseconds
1.025 microseconds for jump taken

OTR — OUTPUT TRANSFER REQUEST (PRIVILEGED)

OTR | OTR

OP CODE:

80

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	0				Relative Buffer End Address							
Function					Mode			Displacement							

DEFINITION:

OTR is a privileged instruction and is only executed as an out-of-sequence instruction in response to a Buffered I/O request from the I/O subsystem. If encountered in a normal program sequence, a no-op is performed and an Instruction Violation (PI) is generated. Normal address calculations are not performed. The displacement field is added to X'0100' and the result is used to obtain the Buffer End Address.

The next contiguous location after the OTR instruction must contain the buffer word count in two's complement. This word is incremented by one if the count is negative and added to the Buffer End Address. The result is used to address memory for the data transfer.

DESIGNATORS:

Zero (D2) - No change
Positive (D3) - No change
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

4.325 microseconds

PJP — POSITIVE JUMP

PJP | PJP

OP CODE:

B0 - B7

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

PJP interrogates the status of bit 3 (Positive Designator) in the D-register. If bit 3 is set, the calculated operand address is transferred to the P-register. If bit 3 is not set, the jump is not executed.

SUMMARY EXPRESSION:

OA P, if D3 = 1

NEXT INSTRUCTION
EXECUTED IF
POSITIVE SET:

Calculated Operand Address + 1

DESIGNATORS:

Zero (D2) - No change
Positive (D3) - No change
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

0.9 microseconds
1.025 microseconds for jump taken

SDA SDA

SUBTRACT DOUBLE-LENGTH ACCUMULATOR

OP CODE:

C8 - CF

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
1 1 0 0 1	m	y, y*
Function	Mode	Displacement

DEFINITION:

SDA algebraically subtracts a double-length operand from the double-length accumulator and leaves the difference in the double-length accumulator.

The double-length operand consists of the operand addressed through the normal calculation sequence and the word contained at the operand location + 1 (least significant half).

Bit 15 of the most significant half of the double word indicates sign: all other bits indicate magnitude.

DESIGNATORS:

Zero	(D2)	-	Set or cleared by difference in accumulator
Positive	(D3)	-	Set or cleared by difference in accumulator
Overflow	(D4)	-	Set if overflow occurs
Carry	(D5)	-	Set or cleared by difference in accumulator

TIMING:

2.6 microseconds

SHF SHF

SHIFT LOCATION

OP CODE:

98 - 9F

FORMAT:

15 14 13 12 11	10 9 8	7 6 5 4 3 2 1 0
1 0 0 1 1	m	y, y*
Function	Mode	Displacement

DEFINITION:

SHF performs a shifting operation on the data word contained in the calculated operand address location. Note: The G-register must be loaded, utilizing the load G (LDG) instruction, prior to the shift execution.

The type of shift operation is specified by the contents of the Shift Description (G) register at location X'0003'. (See Figure 4-1) The shift description word format is:

15	14	13	12 11 10 9 8 7 6 5	4	3	2	1	0
X	X	X		X	X	X	X	X

- Bits 4–0: Shift Count
- Bit 13: Circular or Arithmetic, C/A=1/0
- Bit 14: Right or Left, R/L=1/0
- Bit 15: Double or Single Word, D/S=1/0

For doubleword shifts, the operand address contains the most significant word and the next contiguous location contains the least significant word. The CPU automatically obtains the second word.

The shifting operations specified by bits 13, 14 and 15 of the G-register are shown on the following page. If a shift count of 0 is contained in the G-register, no shift is performed. However, the designators will be set according to the non-shifted value. The no-shift function is valuable for interrogating the contents of a core location. The G-register is unaffected by the shift operation.

DESIGNATORS:

Zero	(D2)	-	Set if result of shift is all zeros
Positive	(D3)	-	Set if most significant bit of result is a ZERO
Overflow	(D4)	-	Set during left shift if sign bit changes
Carry	(D5)	-	Set if bit shifted out of the word (or doubleword) is ONE

TIMING:

Single Word Shift = 2.0 + (74n) microseconds
Doubleword Shift = 2.80 + (1.57n) microseconds
where n = number of shifts

SHIFT LOCATION (Continued)

G-REGISTER BITS 15	14	13	12	SHIFT TYPE	PRE-SHIFTED WORD	ONE SHIFT COUNT
0	0	0	0	Single Left Arithmetic (Zero Insert)	1011001110101111	1 ← 0110011101011110 ← 0
0	0	1	0	Single Left Circular	1011001110101111	0110011101011111
0	1	0	0	Single Right Arithmetic (Sign Extension)	1011001110101111	1 → 1101100111010111 → 1
0	1	1	0	Single Right Circular	1011001110101111	1101100111010111
1	0	0	0	Double Left Arithmetic (Zero Insert)	0110011101011110 1011001110101111	0 ← 1100111010111101 ← 0110011101011110 ← 0
1	0	1	0	Double Left Circular	0110011101011110 1011001110101111	1100111010111101 ← 0110011101011110
1	1	0	0	Double Right Arithmetic (Sign Extension)	0110011101011110 1011001110101111	0 → 0011001110101111 → 0101100111010111 → 1
1	1	1	0	Double Right Circular	0110011101011110 1011001110101111	1011001110101111 → 0101100111010111

STORE STATUS

OP CODE:

E8 - EF (Special Meaning)

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	X	X	X	y							
Function					See Definition			Displacement							

DEFINITION:

SST stores the working registers (D, A, E, G, C, B and P) in descending order, in contiguous locations starting at the operand address.

When bit 10 is ZERO, the calculated address determines the address location for storing the first register of the stack (D-register). Bits 8 and 9, as shown below, determine how the operand address is calculated.

Mode Bits	9	8	Address calculation	
	0	0	Direct, absolute	000-255
	0	1	Indirect, absolute	(000-255)
	1	0	Direct, relative	(P) -128 through +127
	1	1	Indirect, relative	(P) -128 through +127

When bit 10 is ONE, the B index register must contain the address of the first storage location. After the working registers are stored the CPU decrements the contents of the B index register by seven. In this manner, the working registers for nested subroutines may be stored in contiguous locations in memory. The operand address is calculated as shown above to determine the next instruction.

In both cases, (index, not indexed), the next instruction executed after the storing operation is the calculated operand address + 1.

When SST is executed in a normal program sequence in the non-privileged mode (D-register, bit 9=1), bits 8 through 11 (lockout designators) of the Designator Register are not stored.

When SST is executed as an out-of-sequence instruction in response to a Processor Interrupt, bits 12 through 15 (Processor Interrupt Condition Indicators) of the Designator Register are stored, then cleared, and all interrupt lockouts are set. Also, the privileged instruction lockout designator (D-register, bit 9) is reset.

When SST is executed as an out-of-sequence instruction in response to an External Interrupt, the privileged instruction lockout designator (D-register, bit 9) is reset and the External Interrupt lockout (D-register, bit 11) is set.

STORE STATUS (Cont'd)

SUMMARY EXPRESSION:

(P) → OA-6
(B) → OA-5
(C) → OA-4
(G) → OA-3
(E) → OA-2
(A) → OA-1
(D) → OA

NEXT INSTRUCTION EXECUTED:

OA + 1

DESIGNATORS:

The zero, positive, overflow and carry designators are not changed.

TIMING:

9.075 microseconds

STA STA

STORE ACCUMULATOR

OP CODE:

A8 - AF

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

STA transfers the 16-bit word contained in the A (Accumulator) register to the memory location specified by the calculated operand address.

The contents of the accumulator are unchanged.

SUMMARY EXPRESSION:

(A) → OA

DESIGNATORS:

Zero	(D2)	-	Set if word contains all zeros
Positive	(D3)	-	Set if bit 15 of A = 0
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

1.4 microseconds

STE STE

STORE EXTENDED ACCUMULATOR

OP CODE:

A0 - A7

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

STE transfers the 16-bit word contained in the E-register to the memory location specified by the calculated operand address. The contents of E are unchanged.

SUMMARY EXPRESSION:

(E) → OA

DESIGNATORS:

Zero	(D2)	-	Set if word contains all zeros
Positive	(D3)	-	Set if bit 15 of E = 0
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

1.4 microseconds

STOP (PRIVILEGED)

OP CODE:

00 - 07

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	m			y							
Function					Mode			Displacement							

DEFINITION:

STP is a privileged instruction. When executed in the non-privileged mode (Designator Register, bit 9=1), the instruction executes a no-op, sets the Instruction Violation designator (bit 13) in the D-register and generates the Processor Interrupt. The Processor Interrupt causes the CPU to execute the instruction at location X'0103'.

When executed in the privileged mode, the STP instruction performs a normal address calculation, loads the S-register with the operand address and executes a processor halt. All processing and I/O transfers are stopped (with the exception of DMA transfers) until manually restarted at the control panel.

DESIGNATORS:

The designators are unchanged.

TIMING:

0.9 microseconds

STORE ZERO

OP CODE:

38 - 3F

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

STZ clears the memory location specified by the calculated operand address.

SUMMARY EXPRESSION:

$0 \rightarrow OA$

DESIGNATORS:

Zero	(D2)	-	Equal to one
Positive	(D3)	-	Equal to one
Overflow	(D4)	-	No change
Carry	(D5)	-	No change

TIMING:

1.4 microseconds

SUB — SUB

SUBTRACT ACCUMULATOR

OP CODE:

48 - 4F

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

SUB algebraically subtracts the contents of the calculated operand address from the contents of the accumulator. End around carry is not provided, and negative quantities are expressed in two's complement notation.

SUMMARY EXPRESSION:

(A) - (OA) $\rightarrow$ A

DESIGNATORS:

Zero (D2) - Set or cleared as indicated by the remainder in the accumulator.

Positive (D3) - Set or cleared as indicated by the remainder in the accumulator.

Overflow (D4) - Set by a sign change of the accumulator.

Carry (D5) - Set or cleared as indicated by the remainder in the accumulator.

TIMING:

1.7 microseconds

ZJP — ZJP

ZERO JUMP

OP CODE:

F0 - F7

FORMAT:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	m			y, y*							
Function					Mode			Displacement							

DEFINITION:

ZJP interrogates the status of bit 2 (Zero Designator) in the D-register. If bit 2 is set the calculated operand address is transferred to the P-register. If bit 2 is not set, the jump is not executed.

SUMMARY EXPRESSION:

OA $\rightarrow$ P, if D2 = 1

NEXT INSTRUCTION EXECUTED IF ZERO SET:

Calculated Operand Address + 1

DESIGNATORS:

Zero (D2) - No change
Positive (D3) - No change
Overflow (D4) - No change
Carry (D5) - No change

TIMING:

0.9 microseconds
1.025 microseconds for jump taken

APPENDIX

CONVERSION TABLES

This appendix contains the following reference tables:

HEXADECIMAL ARITHMETIC

ADDITION TABLE

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
2	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11
3	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12
4	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
5	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14
6	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15
7	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16
8	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17
9	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18
A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19
B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A
C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

MULTIPLICATION TABLE

1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
3	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
4	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	1E	2B	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

TABLE OF POWERS OF SIXTEEN$_{10}$

16^n	n	16^{-n}
1	0	$0.10000\ 00000\ 00000\ 00000 \times 10$
16	1	$0.62500\ 00000\ 00000\ 00000 \times 10^{-1}$
256	2	$0.39062\ 50000\ 00000\ 00000 \times 10^{-2}$
4 096	3	$0.24414\ 06250\ 00000\ 00000 \times 10^{-3}$
65 536	4	$0.15258\ 78906\ 25000\ 00000 \times 10^{-4}$
1 048 576	5	$0.95367\ 43164\ 06250\ 00000 \times 10^{-6}$
16 777 216	6	$0.59604\ 64477\ 53906\ 25000 \times 10^{-7}$
268 435 456	7	$0.37252\ 90298\ 46191\ 40625 \times 10^{-8}$
4 294 967 296	8	$0.23283\ 06436\ 53869\ 62891 \times 10^{-9}$
68 719 476 736	9	$0.14551\ 91522\ 83668\ 51807 \times 10^{-10}$
1 099 511 627 776	10	$0.90949\ 47017\ 72928\ 23792 \times 10^{-12}$
17 592 186 044 416	11	$0.56843\ 41886\ 08080\ 14870 \times 10^{-13}$
281 474 976 710 656	12	$0.35527\ 13678\ 80050\ 09294 \times 10^{-14}$
4 503 599 627 370 496	13	$0.22204\ 46049\ 25031\ 30808 \times 10^{-15}$
72 057 594 037 927 936	14	$0.13877\ 78780\ 78144\ 56755 \times 10^{-16}$
1 152 921 504 606 846 976	15	$0.86736\ 17379\ 88403\ 54721 \times 10^{-18}$

TABLE OF POWERS OF 10_{16}

10^n	n	10^{-n}
1	0	1.0000 0000 0000 0000
A	1	0.1999 9999 9999 999A
64	2	0.28F5 C28F 5C28 F5C3 × 16^{-1}
3E8	3	0.4189 374B C6A7 EF9E × 16^{-2}
2710	4	0.68DB 8BAC 710C B296 × 16^{-3}
1 86A0	5	0.A7C5 AC47 1B47 8423 × 16^{-4}
F 4240	6	0.10C6 F7A0 B5ED 8D37 × 16^{-4}
98 9680	7	0.1AD7 F29A BCAF 4858 × 16^{-5}
5F5 E100	8	0.2AF3 1DC4 6118 73BF × 16^{-6}
3B9A CA00	9	0.44B8 2FA0 9B5A 52CC × 16^{-7}
2 540B E400	10	0.6DF3 7F67 5EF6 EADF × 16^{-8}
17 4876 E800	11	0.AFEB FF0B CB24 AAFF × 16^{-9}
E8 D4A5 1000	12	0.1197 9981 2DEA 1119 × 16^{-9}
918 4E72 A000	13	0.1C25 C268 4976 81C2 × 16^{-10}
5AF3 107A 4000	14	0.2D09 370D 4257 3604 × 16^{-11}
3 8D7E A4C6 8000	15	0.480E BE7B 9D58 566D × 16^{-12}
23 8652 6FC1 0000	16	0.734A CA5F 6226 F0AE × 16^{-13}
163 4578 5D8A 0000	17	0.B877 AA32 36A4 B449 × 16^{-14}
DE0 B6B3 A764 0000	18	0.1272 5DD1 D243 ABA1 × 16^{-14}
8AC7 2304 89E8 0000	19	0.1D83 C94F B6D2 AC35 × 16^{-15}

HEXADECIMAL-DECIMAL INTEGER CONVERSION

The table below provides for direct conversions between hexadecimal integers in the range 0—FFF and decimal integers in the range 0—4095. For conversion of larger integers, the table values may be added to the following figures:

Hexadecimal	Decimal	Hexadecimal	Decimal
01 000	4 096	20 000	131 072
02 000	8 192	30 000	196 608
03 000	12 288	40 000	262 144
04 000	16 384	50 000	327 680
05 000	20 480	60 000	393 216
06 000	24 576	70 000	458 752
07 000	28 672	80 000	524 288
08 000	32 768	90 000	589 824
09 000	36 864	A0 000	655 360
0A 000	40 960	B0 000	720 896
0B 000	45 056	C0 000	786 432
0C 000	49 152	D0 000	851 968
0D 000	53 248	E0 000	917 504
0E 000	57 344	F0 000	983 040
0F 000	61 440	100 000	1 048 576
10 000	65 536	200 000	2 097 152
11 000	69 632	300 000	3 145 728
12 000	73 728	400 000	4 194 304
13 000	77 824	500 000	5 242 880
14 000	81 920	600 000	6 291 456
15 000	86 016	700 000	7 340 032
16 000	90 112	800 000	8 388 608
17 000	94 208	900 000	9 437 184
18 000	98 304	A00 000	10 485 760
19 000	102 400	B00 000	11 534 336
1A 000	106 496	C00 000	12 582 912
1B 000	110 592	D00 000	13 631 488
1C 000	114 688	E00 000	14 680 064
1D 000	118 784	F00 000	15 728 640
1E 000	122 880	1 000 000	16 777 216
1F 000	126 976	2 000 000	33 554 432

Hexadecimal fractions may be converted to decimal fractions as follows:

1. Express the hexadecimal fraction as an integer times 16^{-n}, where n is the number of significant hexadecimal places to the right of the hexadecimal point.

$$0.CA9BF3_{16} = CA9BF3_{16} \times 16^{-6}$$

2. Find the decimal equivalent of the hexadecimal integer

$$CA9BF3_{16} = 13\,278\,195_{10}$$

3. Multiply the decimal equivalent by 16^{-n}

$$\begin{array}{r} 13\,278\,195 \\ \times\ 596\,046\,448 \times 10^{-16} \\ \hline 0.791\,442\,096_{10} \end{array}$$

Decimal fractions may be converted to hexadecimal fractions by successively multiplying the decimal fraction by 16_{10}. After each multiplication, the integer portion is removed to form a hexadecimal fraction by building to the right of the hexadecimal point. However, since decimal arithmetic is used in this conversion, the integer portion of each product must be converted to hexadecimal numbers.

Example: Convert 0.895_{10} to its hexadecimal equivalent

$$\begin{array}{r} 0.895 \\ 16 \\ \hline 14.320 \\ 16 \\ \hline 5.120 \\ 16 \\ \hline 1.920 \\ 16 \\ \hline 14.720 \end{array}$$

$0.E51E_{16}$

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
010	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
020	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
030	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
040	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
050	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0091	0093	0094	0095
060	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
070	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
080	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
090	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A0	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B0	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C0	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D0	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E0	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F0	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont'd)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
100	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
110	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
120	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
130	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
140	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
150	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
160	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
170	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
180	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
190	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A0	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B0	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C0	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D0	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E0	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F0	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
200	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
210	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
220	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
230	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
240	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
250	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
260	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
270	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
280	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
290	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A0	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B0	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C0	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D0	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E0	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F0	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
300	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
310	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
320	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
330	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
340	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
350	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
360	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
370	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
380	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
390	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A0	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B0	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C0	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D0	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E0	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont'd)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1084	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont'd)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2131	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont'd)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2583	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE0	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF0	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Cont'd)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3649	3650	3651	3652	3653	3654	3655	3656	3567	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB0	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC0	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF0	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F00	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	2983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB0	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD0	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE0	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4706	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

HEXADECIMAL-DECIMAL FRACTION CONVERSION

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.40 00 00 00	.25000 00000	.80 00 00 00	.50000 00000	.C0 00 00 00	.75000 00000
.01 00 00 00	.00390 62500	.41 00 00 00	.25390 62500	.81 00 00 00	.50390 62500	.C1 00 00 00	.75390 62500
.02 00 00 00	.00781 25000	.42 00 00 00	.25781 25000	.82 00 00 00	.50781 25000	.C2 00 00 00	.75781 25000
.03 00 00 00	.01771 87500	.43 00 00 00	.26171 87500	.83 00 00 00	.51171 87500	.C3 00 00 00	.76171 87500
.04 00 00 00	.01562 50000	.44 00 00 00	.26562 50000	.84 00 00 00	.51562 50000	.C4 00 00 00	.76562 50000
.05 00 00 00	.01953 12500	.45 00 00 00	.26953 12500	.85 00 00 00	.51953 12500	.C5 00 00 00	.76953 12500
.06 00 00 00	.02343 75000	.46 00 00 00	.27343 75000	.86 00 00 00	.52343 75000	.C6 00 00 00	.77343 75000
.07 00 00 00	.02734 37500	.47 00 00 00	.27734 37500	.87 00 00 00	.52734 37500	.C7 00 00 00	.77734 37500
.08 00 00 00	.03125 00000	.48 00 00 00	.28125 00000	.88 00 00 00	.53125 00000	.C8 00 00 00	.78125 00000
.09 00 00 00	.03515 62500	.49 00 00 00	.28515 62500	.89 00 00 00	.53515 62500	.C9 00 00 00	.78515 62500
.0A 00 00 00	.03906 25000	.4A 00 00 00	.28906 25000	.8A 00 00 00	.53906 25000	.CA 00 00 00	.78906 25000
.0B 00 00 00	.04296 87500	.4B 00 00 00	.29296 87500	.8B 00 00 00	.54296 87500	.CB 00 00 00	.79296 87500
.0C 00 00 00	.04687 50000	.4C 00 00 00	.29687 50000	.8C 00 00 00	.54687 50000	.CC 00 00 00	.79687 50000
.0D 00 00 00	.05078 12500	.4D 00 00 00	.30078 12500	.8D 00 00 00	.55078 12500	.CD 00 00 00	.80078 12500
.0E 00 00 00	.05468 75000	.4E 00 00 00	.30468 75000	.8E 00 00 00	.55468 75000	.CE 00 00 00	.80468 75000
.0F 00 00 00	.05859 37500	.4F 00 00 00	.30859 37500	.8F 00 00 00	.55859 37500	.CF 00 00 00	.80859 37500
.10 00 00 00	.06250 00000	.50 00 00 00	.31250 00000	.90 00 00 00	.56250 00000	.D0 00 00 00	.81250 00000
.11 00 00 00	.06640 62500	.51 00 00 00	.31640 62500	.91 00 00 00	.56640 62500	.D1 00 00 00	.81640 62500
.12 00 00 00	.07031 25000	.52 00 00 00	.32031 25000	.92 00 00 00	.57031 25000	.D2 00 00 00	.82031 25000
.13 00 00 00	.07421 87500	.53 00 00 00	.32421 87500	.93 00 00 00	.57421 87500	.D3 00 00 00	.82421 87500
.14 00 00 00	.07812 50000	.54 00 00 00	.32812 50000	.94 00 00 00	.57812 50000	.D4 00 00 00	.82812 50000
.15 00 00 00	.08203 12500	.55 00 00 00	.33203 12500	.95 00 00 00	.58203 12500	.D5 00 00 00	.83203 12500
.16 00 00 00	.08593 75000	.56 00 00 00	.33593 75000	.96 00 00 00	.58593 75000	.D6 00 00 00	.83593 75000
.17 00 00 00	.08984 37500	.57 00 00 00	.33984 37500	.97 00 00 00	.58984 37500	.D7 00 00 00	.83984 37500
.18 00 00 00	.09375 00000	.58 00 00 00	.34375 00000	.98 00 00 00	.59375 00000	.D8 00 00 00	.84375 00000
.19 00 00 00	.09765 62500	.59 00 00 00	.34765 62500	.99 00 00 00	.59765 62500	.D9 00 00 00	.84765 62500
.1A 00 00 00	.10156 25000	.5A 00 00 00	.35156 25000	.9A 00 00 00	.60156 25000	.DA 00 00 00	.85156 25000
.1B 00 00 00	.10546 87500	.5B 00 00 00	.35546 87500	.9B 00 00 00	.60546 87500	.DB 00 00 00	.85546 87500
.1C 00 00 00	.10937 50000	.5C 00 00 00	.35937 50000	.9C 00 00 00	.60937 50000	.DC 00 00 00	.85937 50000
.1D 00 00 00	.11328 12500	.5D 00 00 00	.36328 12500	.9D 00 00 00	.61328 12500	.DD 00 00 00	.86328 12500
.1E 00 00 00	.11718 75000	.5E 00 00 00	.36718 75000	.9E 00 00 00	.61718 75000	.DE 00 00 00	.86718 75000
.1F 00 00 00	.12109 37500	.5F 00 00 00	.37109 37500	.9F 00 00 00	.62109 37500	.DF 00 00 00	.87109 37500
.20 00 00 00	.12500 00000	.60 00 00 00	.37500 00000	.A0 00 00 00	.62500 00000	.E0 00 00 00	.87500 00000
.21 00 00 00	.12890 62500	.61 00 00 00	.37890 62500	.A1 00 00 00	.62890 62500	.E1 00 00 00	.87890 62500
.22 00 00 00	.13281 25000	.62 00 00 00	.38281 25000	.A2 00 00 00	.63281 25000	.E2 00 00 00	.88281 25000
.23 00 00 00	.13671 87500	.63 00 00 00	.38671 87500	.A3 00 00 00	.63671 87500	.E3 00 00 00	.88671 87500
.24 00 00 00	.14062 50000	.64 00 00 00	.39062 50000	.A4 00 00 00	.64062 50000	.E4 00 00 00	.89062 50000
.25 00 00 00	.14453 12500	.65 00 00 00	.39453 12500	.A5 00 00 00	.64453 12500	.E5 00 00 00	.89453 12500
.26 00 00 00	.14843 75000	.66 00 00 00	.39843 75000	.A6 00 00 00	.64843 75000	.E6 00 00 00	.89843 75000
.27 00 00 00	.15234 37500	.67 00 00 00	.40234 37500	.A7 00 00 00	.65234 37500	.E7 00 00 00	.90234 37500
.28 00 00 00	.15625 00000	.68 00 00 00	.40625 00000	.A8 00 00 00	.65625 00000	.E8 00 00 00	.90625 00000
.29 00 00 00	.16015 62500	.69 00 00 00	.41015 62500	.A9 00 00 00	.66015 62500	.E9 00 00 00	.91015 62500
.2A 00 00 00	.16406 25000	.6A 00 00 00	.41406 25000	.AA 00 00 00	.66406 25000	.EA 00 00 00	.91406 25000
.2B 00 00 00	.16796 87500	.6B 00 00 00	.41796 87500	.AB 00 00 00	.66796 87500	.EB 00 00 00	.91796 87500
.2C 00 00 00	.17187 50000	.6C 00 00 00	.42187 50000	.AC 00 00 00	.67187 50000	.EC 00 00 00	.92187 50000
.2D 00 00 00	.17578 12500	.6D 00 00 00	.42578 12500	.AD 00 00 00	.67578 12500	.ED 00 00 00	.92578 12500
.2E 00 00 00	.17968 75000	.6E 00 00 00	.42968 75000	.AE 00 00 00	.67968 75000	.EE 00 00 00	.92968 75000
.2F 00 00 00	.18359 37500	.6F 00 00 00	.43359 37500	.AF 00 00 00	.68359 37500	.EF 00 00 00	.93359 37500
.30 00 00 00	.18750 00000	.70 00 00 00	.43750 00000	.B0 00 00 00	.68750 00000	.F0 00 00 00	.93750 00000
.31 00 00 00	.19140 62500	.71 00 00 00	.44140 62500	.B1 00 00 00	.69140 62500	.F1 00 00 00	.94140 62500
.32 00 00 00	.19531 25000	.72 00 00 00	.44531 25000	.B2 00 00 00	.69531 25000	.F2 00 00 00	.94531 25000
.33 00 00 00	.19921 87500	.73 00 00 00	.44921 87500	.B3 00 00 00	.69921 87500	.F3 00 00 00	.94921 87500
.34 00 00 00	.20312 50000	.74 00 00 00	.45312 50000	.B4 00 00 00	.70312 50000	.F4 00 00 00	.95312 50000
.35 00 00 00	.20703 12500	.75 00 00 00	.45703 12500	.B5 00 00 00	.70703 12500	.F5 00 00 00	.95703 12500
.36 00 00 00	.21093 75000	.76 00 00 00	.46093 75000	.B6 00 00 00	.71093 75000	.F6 00 00 00	.96093 75000
.37 00 00 00	.21484 37500	.77 00 00 00	.46484 37500	.B7 00 00 00	.71484 37500	.F7 00 00 00	.96484 37500
.38 00 00 00	.21875 00000	.78 00 00 00	.46875 00000	.B8 00 00 00	.71875 00000	.F8 00 00 00	.96875 00000
.39 00 00 00	.22265 62500	.79 00 00 00	.47265 62500	.B9 00 00 00	.72265 62500	.F9 00 00 00	.97265 62500
.3A 00 00 00	.22656 25000	.7A 00 00 00	.47656 25000	.BA 00 00 00	.72656 25000	.FA 00 00 00	.97656 25000
.3B 00 00 00	.23046 87500	.7B 00 00 00	.48046 87500	.BB 00 00 00	.73046 87500	.FB 00 00 00	.98046 87500
.3C 00 00 00	.23437 50000	.7C 00 00 00	.48437 50000	.BC 00 00 00	.73437 50000	.FC 00 00 00	.98437 50000
.3D 00 00 00	.23828 12500	.7D 00 00 00	.48828 12500	.BD 00 00 00	.73828 12500	.FD 00 00 00	.98828 12500
.3E 00 00 00	.24218 75000	.7E 00 00 00	.49218 75000	.BE 00 00 00	.74218 75000	.FE 00 00 00	.99218 75000
.3F 00 00 00	.24609 37500	.7F 00 00 00	.49609 37500	.BF 00 00 00	.74609 37500	.FF 00 00 00	.99609 37500

HEXADECIMAL-DECIMAL FRACTION CONVERSION (Cont'd)

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 40 00 00	.00097 65625	.00 80 00 00	.00195 31250	.00 C0 00 00	.00292 96875
.00 01 00 00	.00001 52587	.00 41 00 00	.00099 18212	.00 81 00 00	.00196 83837	.00 C1 00 00	.00294 49462
.00 02 00 00	.00003 05175	.00 42 00 00	.00100 70800	.00 82 00 00	.00198 36425	.00 C2 00 00	.00296 02050
.00 03 00 00	.00004 57763	.00 43 00 00	.00102 23388	.00 83 00 00	.00199 89013	.00 C3 00 00	.00297 54638
.00 04 00 00	.00006 10351	.00 44 00 00	.00103 75976	.00 84 00 00	.00201 41601	.00 C4 00 00	.00299 07226
.00 05 00 00	.00007 62939	.00 45 00 00	.00105 28564	.00 85 00 00	.00202 94189	.00 C5 00 00	.00300 59814
.00 06 00 00	.00009 15527	.00 46 00 00	.00106 81152	.00 86 00 00	.00204 46777	.00 C6 00 00	.00302 12402
.00 07 00 00	.00010 68115	.00 47 00 00	.00108 33740	.00 87 00 00	.00205 99365	.00 C7 00 00	.00303 64990
.00 08 00 00	.00012 20703	.00 48 00 00	.00109 86328	.00 88 00 00	.00207 51953	.00 C8 00 00	.00305 17578
.00 09 00 00	.00013 73291	.00 49 00 00	.00111 38916	.00 89 00 00	.00209 04541	.00 C9 00 00	.00306 70166
.00 0A 00 00	.00015 25878	.00 4A 00 00	.00112 91503	.00 8A 00 00	.00210 57128	.00 CA 00 00	.00308 22753
.00 0B 00 00	.00016 78466	.00 4B 00 00	.00114 44091	.00 8B 00 00	.00212 09716	.00 CB 00 00	.00309 75341
.00 0C 00 00	.00018 31504	.00 4C 00 00	.00115 96679	.00 8C 00 00	.00213 62304	.00 CC 00 00	.00311 27929
.00 0D 00 00	.00019 83642	.00 4D 00 00	.00117 49267	.00 8D 00 00	.00215 14892	.00 CD 00 00	.00312 80517
.00 0E 00 00	.00021 36230	.00 4E 00 00	.00119 01855	.00 8E 00 00	.00216 67480	.00 CE 00 00	.00314 33105
.00 0F 00 00	.00022 88818	.00 4F 00 00	.00120 54443	.00 8F 00 00	.00218 20068	.00 CF 00 00	.00315 85693
.00 10 00 00	.00024 41406	.00 50 00 00	.00122 07031	.00 90 00 00	.00219 72656	.00 D0 00 00	.00317 38281
.00 11 00 00	.00025 93994	.00 51 00 00	.00123 59619	.00 91 00 00	.00221 25244	.00 D1 00 00	.00318 90869
.00 12 00 00	.00027 46582	.00 52 00 00	.00125 12207	.00 92 00 00	.00222 77832	.00 D2 00 00	.00320 43457
.00 13 00 00	.00028 99169	.00 53 00 00	.00126 64794	.00 93 00 00	.00224 30419	.00 D3 00 00	.00321 96044
.00 14 00 00	.00030 51757	.00 54 00 00	.00128 17382	.00 94 00 00	.00225 83007	.00 D4 00 00	.00323 48362
.00 15 00 00	.00032 04345	.00 55 00 00	.00129 69970	.00 95 00 00	.00227 35595	.00 D5 00 00	.00325 01220
.00 16 00 00	.00033 56933	.00 56 00 00	.00131 22558	.00 96 00 00	.00228 88183	.00 D6 00 00	.00326 53808
.00 17 00 00	.00035 09521	.00 57 00 00	.00132 75146	.00 97 00 00	.00230 40771	.00 D7 00 00	.00328 06396
.00 18 00 00	.00036 62109	.00 58 00 00	.00134 27734	.00 98 00 00	.00231 93359	.00 D8 00 00	.00329 58984
.00 19 00 00	.00038 14697	.00 59 00 00	.00135 80322	.00 99 00 00	.00233 45947	.00 D9 00 00	.00331 11572
.00 1A 00 00	.00039 67285	.00 5A 00 00	.00137 32910	.00 9A 00 00	.00234 98535	.00 DA 00 00	.00332 64160
.00 1B 00 00	.00041 19873	.00 5B 00 00	.00138 85498	.00 9B 00 00	.00236 51123	.00 DB 00 00	.00334 16748
.00 1C 00 00	.00042 72460	.00 5C 00 00	.00140 38085	.00 9C 00 00	.00238 03710	.00 DC 00 00	.00335 69335
.00 1D 00 00	.00044 25048	.00 5D 00 00	.00141 90673	.00 9D 00 00	.00239 56298	.00 DD 00 00	.00337 21923
.00 1E 00 00	.00045 77636	.00 5E 00 00	.00143 43261	.00 9E 00 00	.00241 08886	.00 DE 00 00	.00338 74511
.00 1F 00 00	.00047 30224	.00 5F 00 00	.00144 95849	.00 9F 00 00	.00242 61474	.00 DF 00 00	.00340 27099
.00 20 00 00	.00048 82812	.00 60 00 00	.00146 48437	.00 A0 00 00	.00244 14062	.00 E0 00 00	.00341 79687
.00 21 00 00	.00050 35400	.00 61 00 00	.00148 01025	.00 A1 00 00	.00245 66650	.00 E1 00 00	.00343 32275
.00 22 00 00	.00051 87988	.00 62 00 00	.00149 53613	.00 A2 00 00	.00247 19238	.00 E2 00 00	.00344 84863
.00 23 00 00	.00053 40576	.00 63 00 00	.00151 06201	.00 A3 00 00	.00248 71826	.00 E3 00 00	.00346 37451
.00 24 00 00	.00054 93164	.00 64 00 00	.00152 58789	.00 A4 00 00	.00250 24414	.00 E4 00 00	.00347 90039
.00 25 00 00	.00056 45751	.00 65 00 00	.00154 11376	.00 A5 00 00	.00251 77001	.00 E5 00 00	.00349 42626
.00 26 00 00	.00057 98339	.00 66 00 00	.00155 63964	.00 A6 00 00	.00253 29589	.00 E6 00 00	.00350 95214
.00 27 00 00	.00059 50927	.00 67 00 00	.00157 16552	.00 A7 00 00	.00254 82177	.00 E7 00 00	.00352 47802
.00 28 00 00	.00061 03515	.00 68 00 00	.00158 69140	.00 A8 00 00	.00256 34765	.00 E8 00 00	.00354 00390
.00 29 00 00	.00062 56103	.00 69 00 00	.00160 21728	.00 A9 00 00	.00257 87353	.00 E9 00 00	.00355 52978
.00 2A 00 00	.00064 08691	.00 6A 00 00	.00161 74316	.00 AA 00 00	.00259 39941	.00 EA 00 00	.00357 05566
.00 2B 00 00	.00065 61279	.00 6B 00 00	.00163 26904	.00 AB 00 00	.00260 92529	.00 EB 00 00	.00358 58154
.00 2C 00 00	.00067 13867	.00 6C 00 00	.00164 79492	.00 AC 00 00	.00262 45117	.00 EC 00 00	.00360 10742
.00 2D 00 00	.00068 66455	.00 6D 00 00	.00166 32080	.00 AD 00 00	.00263 97705	.00 ED 00 00	.00361 63330
.00 2E 00 00	.00070 19042	.00 6E 00 00	.00167 84667	.00 AE 00 00	.00265 50292	.00 EE 00 00	.00363 15917
.00 2F 00 00	.00071 71630	.00 6F 00 00	.00169 37255	.00 AF 00 00	.00267 02880	.00 EF 00 00	.00364 68505
.00 30 00 00	.00073 24218	.00 70 00 00	.00170 89843	.00 B0 00 00	.00268 55468	.00 F0 00 00	.00366 21093
.00 31 00 00	.00074 76806	.00 71 00 00	.00172 42431	.00 B1 00 00	.00270 08056	.00 F1 00 00	.00367 73681
.00 32 00 00	.00076 29394	.00 72 00 00	.00173 95019	.00 B2 00 00	.02271 60644	.00 F2 00 00	.00369 26269
.00 33 00 00	.00077 81982	.00 73 00 00	.00175 47607	.00 B3 00 00	.00273 13232	.00 F3 00 00	.00370 78857
.00 34 00 00	.00079 34570	.00 74 00 00	.00177 00195	.00 B4 00 00	.00274 65820	.00 F4 00 00	.00372 31445
.00 35 00 00	.00080 87158	.00 75 00 00	.00178 52783	.00 B5 00 00	.00276 18408	.00 F5 00 00	.00373 84033
.00 36 00 00	.00082 39746	.00 76 00 00	.00180 05371	.00 B6 00 00	.00277 70996	.00 F6 00 00	.00375 36621
.00 37 00 00	.00083 92333	.00 77 00 00	.00181 57958	.00 B7 00 00	.00279 23583	.00 F7 00 00	.00376 89208
.00 38 00 00	.00085 44921	.00 78 00 00	.00183 10546	.00 B8 00 00	.00280 76171	.00 F8 00 00	.00378 41796
.00 39 00 00	.00086 97509	.00 79 00 00	.00184 63134	.00 B9 00 00	.00282 28759	.00 F9 00 00	.00379 94384
.00 3A 00 00	.00088 50097	.00 7A 00 00	.00186 15722	.00 BA 00 00	.00283 81347	.00 FA 00 00	.00381 46972
.00 3B 00 00	.00090 02685	.00 7B 00 00	.00187 68310	.00 BB 00 00	.00285 33935	.00 FB 00 00	.00382 99560
.00 3C 00 00	.00091 55273	.00 7C 00 00	.00189 20898	.00 BC 00 00	.00286 86523	.00 FC 00 00	.00384 52148
.00 3D 00 00	.00093 07861	.00 7D 00 00	.00190 73486	.00 BD 00 00	.00288 39111	.00 FD 00 00	.00386 04736
.00 3E 00 00	.00094 60449	.00 7E 00 00	.00192 26074	.00 BE 00 00	.00289 91699	.00 FE 00 00	.00387 57324
.00 3F 00 00	.00096 13037	.00 7F 00 00	.00193 78662	.00 BF 00 00	.00291 44287	.00 FF 00 00	.00389 09912

HEXADECIMAL-DECIMAL FRACTION CONVERSION (Cont'd)

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 00 40 00	.00000 38146	.00 00 80 00	.00000 76293	.00 00 C0 00	.00001 14440
.00 00 01 00	.00000 00596	.00 00 41 00	.00000 38743	.00 00 81 00	.00000 76889	.00 00 C1 00	.00001 15036
.00 00 02 00	.00000 01192	.00 00 42 00	.00000 39339	.00 00 82 00	.00000 77486	.00 00 C2 00	.00001 15633
.00 00 03 00	.00000 01788	.00 00 43 00	.00000 39935	.00 00 83 00	.00000 78082	.00 00 C3 00	.00001 16229
.00 00 04 00	.00000 02384	.00 00 44 00	.00000 40531	.00 00 84 00	.00000 78678	.00 00 C4 00	.00001 16825
.00 00 05 00	.00000 02980	.00 00 45 00	.00000 41127	.00 00 85 00	.00000 79274	.00 00 C5 00	.00001 17421
.00 00 06 00	.00000 03576	.00 00 46 00	.00000 41723	.00 00 86 00	.00000 79870	.00 00 C6 00	.00001 18017
.00 00 07 00	.00000 04172	.00 00 47 00	.00000 42319	.00 00 87 00	.00000 80466	.00 00 C7 00	.00001 18613
.00 00 08 00	.00000 04768	.00 00 48 00	.00000 42915	.00 00 88 00	.00000 81062	.00 00 C8 00	.00001 19209
.00 00 09 00	.00000 05364	.00 00 49 00	.00000 43511	.00 00 89 00	.00000 81658	.00 00 C9 00	.00001 19805
.00 00 0A 00	.00000 05960	.00 00 4A 00	.00000 44107	.00 00 8A 00	.00000 82254	.00 00 CA 00	.00001 20401
.00 00 0B 00	.00000 06556	.00 00 4B 00	.00000 44703	.00 00 8B 00	.00000 82850	.00 00 CB 00	.00001 20997
.00 00 0C 00	.00000 07152	.00 00 4C 00	.00000 45299	.00 00 8C 00	.00000 83446	.00 00 CC 00	.00001 21593
.00 00 0D 00	.00000 07748	.00 00 4D 00	.00000 45895	.00 00 8D 00	.00000 84042	.00 00 CD 00	.00001 22189
.00 00 0E 00	.00000 08344	.00 00 4E 00	.00000 46491	.00 00 8E 00	.00000 84638	.00 00 CE 00	.00001 22785
.00 00 0F 00	.00000 08940	.00 00 4F 00	.00000 47087	.00 00 8F 00	.00000 85234	.00 00 CF 00	.00001 23381
.00 00 10 00	.00000 09536	.00 00 50 00	.00000 47683	.00 00 90 00	.00000 85830	.00 00 D0 00	.00001 23977
.00 00 11 00	.00000 10132	.00 00 51 00	.00000 48279	.00 00 91 00	.00000 86426	.00 00 D1 00	.00001 24573
.00 00 12 00	.00000 10728	.00 00 52 00	.00000 48875	.00 00 92 00	.00000 87022	.00 00 D2 00	.00001 25169
.00 00 13 00	.00000 11324	.00 00 53 00	.00000 49471	.00 00 93 00	.00000 87618	.00 00 D3 00	.00001 25765
.00 00 14 00	.00000 11920	.00 00 54 00	.00000 50067	.00 00 94 00	.00000 88214	.00 00 D4 00	.00001 26361
.00 00 15 00	.00000 12516	.00 00 55 00	.00000 50663	.00 00 95 00	.00000 88810	.00 00 D5 00	.00001 26957
.00 00 16 00	.00000 13113	.00 00 56 00	.00000 51259	.00 00 96 00	.00000 89406	.00 00 D6 00	.00001 27553
.00 00 17 00	.00000 13709	.00 00 57 00	.00000 51856	.00 00 97 00	.00000 90003	.00 00 D7 00	.00001 28149
.00 00 18 00	.00000 14305	.00 00 58 00	.00000 52452	.00 00 98 00	.00000 90599	.00 00 D8 00	.00001 28746
.00 00 19 00	.00000 14901	.00 00 59 00	.00000 53048	.00 00 99 00	.00000 91195	.00 00 D9 00	.00001 29342
.00 00 1A 00	.00000 15497	.00 00 5A 00	.00000 53644	.00 00 9A 00	.00000 91791	.00 00 DA 00	.00001 29938
.00 00 1B 00	.00000 16093	.00 00 5B 00	.00000 54240	.00 00 9B 00	.00000 92387	.00 00 DB 00	.00001 30534
.00 00 1C 00	.00000 16689	.00 00 5C 00	.00000 54836	.00 00 9C 00	.00000 92983	.00 00 DC 00	.00001 31130
.00 00 1D 00	.00000 17285	.00 00 5D 00	.00000 55432	.00 00 9D 00	.00000 93579	.00 00 DD 00	.00001 31726
.00 00 1E 00	.00000 17881	.00 00 5E 00	.00000 56028	.00 00 9E 00	.00000 94175	.00 00 DE 00	.00001 32322
.00 00 1F 00	.00000 18477	.00 00 5F 00	.00000 56624	.00 00 9F 00	.00000 94771	.00 00 DF 00	.00001 32918
.00 00 20 00	.00000 19073	.00 00 60 00	.00000 57220	.00 00 A0 00	.00000 95367	.00 00 E0 00	.00001 33514
.00 00 21 00	.00000 19669	.00 00 61 00	.00000 57816	.00 00 A1 00	.00000 95963	.00 00 E1 00	.00001 34110
.00 00 22 00	.00000 20265	.00 00 62 00	.00000 58412	.00 00 A2 00	.00000 96559	.00 00 E2 00	.00001 34706
.00 00 23 00	.00000 20861	.00 00 63 00	.00000 59008	.00 00 A3 00	.00000 97155	.00 00 E3 00	.00001 35302
.00 00 24 00	.00000 21457	.00 00 64 00	.00000 59604	.00 00 A4 00	.00000 97751	.00 00 E4 00	.00001 35898
.00 00 25 00	.00000 22053	.00 00 65 00	.00000 60200	.00 00 A5 00	.00000 98347	.00 00 E5 00	.00001 36494
.00 00 26 00	.00000 22649	.00 00 66 00	.00000 60796	.00 00 A6 00	.00000 98943	.00 00 E6 00	.00001 37090
.00 00 27 00	.00000 23245	.00 00 67 00	.00000 61392	.00 00 A7 00	.00000 99539	.00 00 E7 00	.00001 37686
.00 00 28 00	.00000 23841	.00 00 68 00	.00000 61988	.00 00 A8 00	.00001 00135	.00 00 E8 00	.00001 38282
.00 00 29 00	.00000 24437	.00 00 69 00	.00000 62584	.00 00 A9 00	.00001 00731	.00 00 E9 00	.00001 38878
.00 00 2A 00	.00000 25033	.00 00 6A 00	.00000 63180	.00 00 AA 00	.00001 01327	.00 00 EA 00	.00001 39474
.00 00 2B 00	.00000 25629	.00 00 6B 00	.00000 63776	.00 00 AB 00	.00001 01923	.00 00 EB 00	.00001 40070
.00 00 2C 00	.00000 26226	.00 00 6C 00	.00000 64373	.00 00 AC 00	.00001 02519	.00 00 EC 00	.00001 40666
.00 00 2D 00	.00000 26822	.00 00 6D 00	.00000 64969	.00 00 AD 00	.00001 03116	.00 00 ED 00	.00001 41263
.00 00 2E 00	.00000 27418	.00 00 6E 00	.00000 65565	.00 00 AE 00	.00001 03712	.00 00 EE 00	.00001 41859
.00 00 2F 00	.00000 28014	.00 00 6F 00	.00000 66161	.00 00 AF 00	.00001 04308	.00 00 EF 00	.00001 42455
.00 00 30 00	.00000 28610	.00 00 70 00	.00000 66757	.00 00 B0 00	.00001 04904	.00 00 F0 00	.00001 43051
.00 00 31 00	.00000 29206	.00 00 71 00	.00000 67353	.00 00 B1 00	.00001 05500	.00 00 F1 00	.00001 43647
.00 00 32 00	.00000 29802	.00 00 72 00	.00000 67949	.00 00 B2 00	.00001 06096	.00 00 F2 00	.00001 44243
.00 00 33 00	.00000 30398	.00 00 73 00	.00000 68545	.00 00 B3 00	.00001 06692	.00 00 F3 00	.00001 44839
.00 00 34 00	.00000 30994	.00 00 74 00	.00000 69141	.00 00 B4 00	.00001 07288	.00 00 F4 00	.00001 45435
.00 00 35 00	.00000 31590	.00 00 75 00	.00000 69737	.00 00 B5 00	.00001 07884	.00 00 F5 00	.00001 46031
.00 00 36 00	.00000 32186	.00 00 76 00	.00000 70333	.00 00 B6 00	.00001 08480	.00 00 F6 00	.00001 46627
.00 00 37 00	.00000 32782	.00 00 77 00	.00000 70929	.00 00 B7 00	.00001 09076	.00 00 F7 00	.00001 47223
.00 00 38 00	.00000 33378	.00 00 78 00	.00000 71525	.00 00 B8 00	.00001 09672	.00 00 F8 00	.00001 47819
.00 00 39 00	.00000 33974	.00 00 79 00	.00000 72121	.00 00 B9 00	.00001 10268	.00 00 F9 00	.00001 48415
.00 00 3A 00	.00000 34570	.00 00 7A 00	.00000 72717	.00 00 BA 00	.00001 10864	.00 00 FA 00	.00001 49011
.00 00 3B 00	.00000 35166	.00 00 7B 00	.00000 73313	.00 00 BB 00	.00001 11460	.00 00 FB 00	.00001 49607
.00 00 3C 00	.00000 35762	.00 00 7C 00	.00000 73909	.00 00 BC 00	.00001 12056	.00 00 FC 00	.00001 50203
.00 00 3D 00	.00000 36358	.00 00 7D 00	.00000 74505	.00 00 BD 00	.00001 12652	.00 00 FD 00	.00001 50799
.00 00 3E 00	.00000 36954	.00 00 7E 00	.00000 75101	.00 00 BE 00	.00001 13248	.00 00 FE 00	.00001 51395
.00 00 3F 00	.00000 37550	.00 00 7F 00	.00000 75697	.00 00 BF 00	.00001 13844	.00 00 FF 00	.00001 51991

HEXADECIMAL-DECIMAL FRACTION CONVERSION (Cont'd)

Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal	Hexadecimal	Decimal
.00 00 00 00	.00000 00000	.00 00 00 40	.00000 00149	.00 00 00 80	.00000 00298	.00 00 00 C0	.00000 00447
.00 00 00 01	.00000 00002	.00 00 00 41	.00000 00151	.00 00 00 81	.00000 00300	.00 00 00 C1	.00000 00449
.00 00 00 02	.00000 00004	.00 00 00 42	.00000 00153	.00 00 00 82	.00000 00302	.00 00 00 C2	.00000 00451
.00 00 00 03	.00000 00006	.00 00 00 43	.00000 00155	.00 00 00 83	.00000 00305	.00 00 00 C3	.00000 00454
.00 00 00 04	.00000 00009	.00 00 00 44	.00000 00158	.00 00 00 84	.00000 00307	.00 00 00 C4	.00000 00456
.00 00 00 05	.00000 00011	.00 00 00 45	.00000 00160	.00 00 00 85	.00000 00309	.00 00 00 C5	.00000 00458
.00 00 00 06	.00000 00013	.00 00 00 46	.00000 00162	.00 00 00 86	.00000 00311	.00 00 00 C6	.00000 00461
.00 00 00 07	.00000 00016	.00 00 00 47	.00000 00165	.00 00 00 87	.00000 00314	.00 00 00 C7	.00000 00463
.00 00 00 08	.00000 00018	.00 00 00 48	.00000 00167	.00 00 00 88	.00000 00316	.00 00 00 C8	.00000 00465
.00 00 00 09	.00000 00020	.00 00 00 49	.00000 00169	.00 00 00 89	.00000 00318	.00 00 00 C9	.00000 00467
.00 00 00 0A	.00000 00023	.00 00 00 4A	.00000 00172	.00 00 00 8A	.00000 00321	.00 00 00 CA	.00000 00470
.00 00 00 0B	.00000 00025	.00 00 00 4B	.00000 00174	.00 00 00 8B	.00000 00323	.00 00 00 CB	.00000 00472
.00 00 00 0C	.00000 00027	.00 00 00 4C	.00000 00176	.00 00 00 BC	.00000 00325	.00 00 00 CC	.00000 00474
.00 00 00 0D	.00000 00030	.00 00 00 4D	.00000 00179	.00 00 00 8D	.00000 00328	.00 00 00 CD	.00000 00477
.00 00 00 0E	.00000 00032	.00 00 00 4E	.00000 00181	.00 00 00 8E	.00000 00330	.00 00 00 CE	.00000 00479
.00 00 00 0F	.00000 00034	.00 00 00 4F	.00000 00183	.00 00 00 8F	.00000 00332	.00 00 00 CF	.00000 00481
.00 00 00 10	.00000 00037	.00 00 00 50	.00000 00186	.00 00 00 90	.00000 00335	.00 00 00 D0	.00000 00484
.00 00 00 11	.00000 00039	.00 00 00 51	.00000 00188	.00 00 00 91	.00000 00337	.00 00 00 D1	.00000 00486
.00 00 00 12	.00000 00041	.00 00 00 52	.00000 00190	.00 00 00 92	.00000 00339	.00 00 00 D2	.00000 00488
.00 00 00 13	.00000 00044	.00 00 00 53	.00000 00193	.00 00 00 93	.00000 00342	.00 00 00 D3	.00000 00491
.00 00 00 14	.00000 00046	.00 00 00 54	.00000 00195	.00 00 00 94	.00000 00344	.00 00 00 D4	.00000 00493
.00 00 00 15	.00000 00048	.00 00 00 55	.00000 00197	.00 00 00 95	.00000 00346	.00 00 00 D5	.00000 00495
.00 00 00 16	.00000 00051	.00 00 00 56	.00000 00200	.00 00 00 96	.00000 00349	.00 00 00 D6	.00000 00498
.00 00 00 17	.00000 00053	.00 00 00 57	.00000 00202	.00 00 00 97	.00000 00351	.00 00 00 D7	.00000 00500
.00 00 00 18	.00000 00055	.00 00 00 58	.00000 00204	.00 00 00 98	.00000 00353	.00 00 00 D8	.00000 00502
.00 00 00 19	.00000 00058	.00 00 00 59	.00000 00207	.00 00 00 99	.00000 00356	.00 00 00 D9	.00000 00505
.00 00 00 1A	.00000 00060	.00 00 00 5A	.00000 00209	.00 00 00 9A	.00000 00358	.00 00 00 DA	.00000 00507
.00 00 00 1B	.00000 00062	.00 00 00 5B	.00000 00211	.00 00 00 9B	.00000 00360	.00 00 00 DB	.00000 00509
.00 00 00 1C	.00000 00065	.00 00 00 5C	.00000 00214	.00 00 00 9C	.00000 00363	.00 00 00 DC	.00000 00512
.00 00 00 1D	.00000 00067	.00 00 00 5D	.00000 00216	.00 00 00 9D	.00000 00365	.00 00 00 DD	.00000 00514
.00 00 00 1E	.00000 00069	.00 00 00 5E	.00000 00218	.00 00 00 9E	.00000 00367	.00 00 00 DE	.00000 00516
.00 00 00 1F	.00000 00072	.00 00 00 5F	.00000 00221	.00 00 00 9F	.00000 00370	.00 00 00 DF	.00000 00519
.00 00 00 20	.00000 00074	.00 00 00 60	.00000 00223	.00 00 00 A0	.00000 00372	.00 00 00 E0	.00000 00521
.00 00 00 21	.00000 00076	.00 00 00 61	.00000 00225	.00 00 00 A1	.00000 00374	.00 00 00 E1	.00000 00523
.00 00 00 22	.00000 00079	.00 00 00 62	.00000 00228	.00 00 00 A2	.00000 00377	.00 00 00 E2	.00000 00526
.00 00 00 23	.00000 00081	.00 00 00 63	.00000 00230	.00 00 00 A3	.00000 00379	.00 00 00 E3	.00000 00528
.00 00 00 24	.00000 00083	.00 00 00 64	.00000 00232	.00 00 00 A4	.00000 00381	.00 00 00 E4	.00000 00530
.00 00 00 25	.00000 00086	.00 00 00 65	.00000 00235	.00 00 00 A5	.00000 00384	.00 00 00 E5	.00000 00533
.00 00 00 26	.00000 00088	.00 00 00 66	.00000 00237	.00 00 00 A6	.00000 00386	.00 00 00 E6	.00000 00535
.00 00 00 27	.00000 00090	.00 00 00 67	.00000 00239	.00 00 00 A7	.00000 00388	.00 00 00 E7	.00000 00537
.00 00 00 28	.00000 00093	.00 00 00 68	.00000 00242	.00 00 00 A8	.00000 00391	.00 00 00 E8	.00000 00540
.00 00 00 29	.00000 00095	.00 00 00 69	.00000 00244	.00 00 00 A9	.00000 00393	.00 00 00 E9	.00000 00542
.00 00 00 2A	.00000 00097	.00 00 00 6A	.00000 00246	.00 00 00 AA	.00000 00395	.00 00 00 EA	.00000 00544
.00 00 00 2B	.00000 00100	.00 00 00 6B	.00000 00249	.00 00 00 AB	.00000 00398	.00 00 00 EB	.00000 00547
.00 00 00 2C	.00000 00102	.00 00 00 6C	.00000 00251	.00 00 00 AC	.00000 00400	.00 00 00 EC	.00000 00549
.00 00 00 2D	.00000 00104	.00 00 00 6D	.00000 00253	.00 00 00 AD	.00000 00402	.00 00 00 ED	.00000 00551
.00 00 00 2E	.00000 00107	.00 00 00 6E	.00000 00256	.00 00 00 AE	.00000 00405	.00 00 00 EE	.00000 00554
.00 00 00 2F	.00000 00109	.00 00 00 6F	.00000 00258	.00 00 00 AF	.00000 00407	.00 00 00 EF	.00000 00556
.00 00 00 30	.00000 00111	.00 00 00 70	.00000 00260	.00 00 00 B0	.00000 00409	.00 00 00 F0	.00000 00558
.00 00 00 31	.00000 00114	.00 00 00 71	.00000 00263	.00 00 00 B1	.00000 00412	.00 00 00 F1	.00000 00561
.00 00 00 32	.00000 00116	.00 00 00 72	.00000 00265	.00 00 00 B2	.00000 00414	.00 00 00 F2	.00000 00563
.00 00 00 33	.00000 00118	.00 00 00 73	.00000 00267	.00 00 00 B3	.00000 00416	.00 00 00 F3	.00000 00565
.00 00 00 34	.00000 00121	.00 00 00 74	.00000 00270	.00 00 00 B4	.00000 00419	.00 00 00 F4	.00000 00568
.00 00 00 35	.00000 00123	.00 00 00 75	.00000 00272	.00 00 00 B5	.00000 00421	.00 00 00 F5	.00000 00570
.00 00 00 36	.00000 00125	.00 00 00 76	.00000 00274	.00 00 00 B6	.00000 00423	.00 00 00 F6	.00000 00572
.00 00 00 37	.00000 00128	.00 00 00 77	.00000 00277	.00 00 00 B7	.00000 00426	.00 00 00 F7	.00000 00575
.00 00 00 38	.00000 00130	.00 00 00 78	.00000 00279	.00 00 00 B8	.00000 00428	.00 00 00 F8	.00000 00577
.00 00 00 39	.00000 00132	.00 00 00 79	.00000 00281	.00 00 00 B9	.00000 00430	.00 00 00 F9	.00000 00579
.00 00 00 3A	.00000 00135	.00 00 00 7A	.00000 00284	.00 00 00 BA	.00000 00433	.00 00 00 FA	.00000 00582
.00 00 00 3B	.00000 00137	.00 00 00 7B	.00000 00286	.00 00 00 BB	.00000 00435	.00 00 00 FB	.00000 00584
.00 00 00 3C	.00000 00139	.00 00 00 7C	.00000 00288	.00 00 00 BC	.00000 00437	.00 00 00 FC	.00000 00586
.00 00 00 3D	.00000 00142	.00 00 00 7D	.00000 00291	.00 00 00 BD	.00000 00440	.00 00 00 FD	.00000 00589
.00 00 00 3E	.00000 00144	.00 00 00 7E	.00000 00293	.00 00 00 BE	.00000 00442	.00 00 00 FE	.00000 00591
.00 00 00 3F	.00000 00146	.00 00 00 7F	.00000 00295	.00 00 00 BF	.00000 00444	.00 00 00 FF	.00000 00593

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125
1 099 511 627 776	40	0.000 000 000 000 909 494 701 772 928 237 915 039 062 5
2 199 023 255 552	41	0.000 000 000 000 454 747 350 886 464 118 957 519 531 25
4 398 046 511 104	42	0.000 000 000 000 227 373 675 443 232 059 478 759 765 625
8 796 093 022 208	43	0.000 000 000 000 113 686 837 721 616 029 739 379 882 812 5
17 592 186 044 416	44	0.000 000 000 000 056 843 418 860 808 014 869 689 941 406 25
35 184 372 088 832	45	0.000 000 000 000 028 421 709 430 404 007 434 844 970 703 125
70 368 744 177 664	46	0.000 000 000 000 014 210 854 715 202 003 717 422 485 351 562 5
140 737 488 355 328	47	0.000 000 000 000 007 105 427 357 601 001 858 711 242 675 781 25
281 474 976 710 656	48	0.000 000 000 000 003 552 713 678 800 500 929 355 621 337 890 625
562 949 953 421 312	49	0.000 000 000 000 001 776 356 839 400 250 464 677 810 668 945 312 5
1 125 899 906 842 624	50	0.000 000 000 000 000 888 178 419 700 125 232 338 905 334 472 656 25
2 251 799 813 685 248	51	0.000 000 000 000 000 444 089 209 850 062 616 169 452 667 236 328 125
4 503 599 627 370 496	52	0.000 000 000 000 000 222 044 604 925 031 308 084 726 333 618 164 062 5
9 007 199 254 740 992	53	0.000 000 000 000 000 111 022 302 462 515 654 042 363 166 809 082 031 25
18 014 398 509 481 984	54	0.000 000 000 000 000 055 511 151 231 257 827 021 181 583 404 541 015 625
36 028 797 018 963 968	55	0.000 000 000 000 000 027 755 575 615 628 913 510 590 791 702 270 507 812 5
72 057 594 037 927 936	56	0.000 000 000 000 000 013 877 787 807 814 456 755 295 395 851 135 253 906 25
144 115 188 075 855 872	57	0.000 000 000 000 000 006 938 893 903 907 228 377 647 697 925 567 626 953 125
288 230 376 151 711 744	58	0.000 000 000 000 000 003 469 446 951 953 614 188 823 848 962 783 813 476 562 5
576 460 752 303 423 488	59	0.000 000 000 000 000 001 734 723 475 976 807 094 411 924 481 391 906 738 281 25
1 152 921 504 606 846 976	60	0.000 000 000 000 000 000 867 361 737 988 403 547 205 962 240 695 953 369 140 625
2 305 843 009 213 693 952	61	0.000 000 000 000 000 000 433 680 868 994 201 773 602 981 120 347 976 684 570 312 5
4 611 686 018 427 387 904	62	0.000 000 000 000 000 000 216 840 434 497 100 886 801 490 560 173 988 342 285 156 25
9 223 372 036 854 775 808	63	0.000 000 000 000 000 000 108 420 217 248 550 443 400 745 280 086 994 171 142 578 125

MATHEMATICAL CONSTANTS

Constant	Decimal Value	Hexadecimal Value
π	3.14159 26535 89793	3.243F 6A89
π^{-1}	0.31830 98861 83790	0.517C C1B7
$\sqrt{\pi}$	1.77245 38509 05516	1.C5BF 891C
$\ln \pi$	1.14472 98858 49400	1.250D 048F
e	1.71828 18284 59045	2.B7E1 5163
e^{-1}	0.36787 94411 71442	0.5E2D 58D9
$\sqrt{e}$	1.64872 12707 00128	1.A612 98E2
$\log_{10} e$	0.43429 44819 03252	0.6F2D EC55
$\log_2 e$	1.44269 50408 88963	1.7154 7653
γ	0.57721 56649 01533	0.93C4 67E4
$\ln \gamma$	-0.54953 93129 81645	-0.8CAE 9BC1
$\sqrt{2}$	1.41421 35623 73095	1.6A09 E668
$\ln 2$	0.69314 71805 59945	0.B172 17F8
$\log_{10} 2$	0.30102 99956 63981	0.4D10 4D42
$\sqrt{10}$	3.16227 76601 68379	3.298B 075C
$\ln 10$	2.30258 40929 94046	2.4D76 3777

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From ______________________ Date ______________________

Title ______________________

Company ______________________ Publication Title ______________________

Address ______________________ Publication Number ______________________

____

____

FOLD FOLD

Check the appropriate item.

Error (Page No. ____, Drawing No. ________)

Addition (Page No. ____, Drawing No. ________)

Other (Page No. ____, Drawing No. ________)

Explanation:

CUT ALONG LINE

FOLD FOLD

FOLD AND STAPLE

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

POSTAGE
WILL BE PAID
BY
ADDRESSEE

NO
POSTAGE STAMP
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

PERMIT NUMBER 4991, ORLANDO, FLORIDA

Customer Support Group
WESTINGHOUSE COMPUTER DEPARTMENT
Westinghouse Electric Corporation
1200 West Colonial Drive
Orlando, Florida 32804

WESTINGHOUSE 2500 INSTRUCTION SET IN HEXADECIMAL

INSTRUCTION	ABSOLUTE		RELATIVE		PRE-INDEX B		PRE-INDEX C	
	DIR	IND	DIR	IND	DIR	IND	DIR	IND
STP	00	01	02	03	04	05	06	07
LDB	08	09	0A	0B	0C	0D	0E	0F
LDC	10	11	12	13	14	15	16	17
LDG	18	19	1A	1B	1C	1D	1E	1F
LDE	20	21	22	23	24	25	26	27
LDA	28	29	2A	2B	2C	2D	2E	2F
CDR	30*	31*	32*	33*	34*	35*	36*	37*
STZ	38	39	3A	3B	3C	3D	3E	3F
ADD	40	41	42	43	44	45	46	47
SUB	48	49	4A	4B	4C	4D	4E	4F
EOR	50	51	52	53	54	55	56	57
AND	58	59	5A	5B	5C	5D	5E	5F
INC	60	61	62	63	64	65	66	67
DCR	68	69	6A	6B	6C	6D	6E	6F
JMP	70	71	72	73	74	75	76	77
OJP	78	79	7A	7B	7C	7D	7E	7F
OTR	80*							
IOA	88*	89*	8A*	8B*	8C*	8D*	8E*	8F*
ITR	90*							
SHF	98	99	9A	9B	9C	9D	9E	9F
STL	A0	A1	A2	A3	A4	A5	A6	A7
STA	A8	A9	AA	AB	AC	AD	AE	AF
PJP	B0	B1	B2	B3	B4	B5	B6	B7
NJP	B8	B9	BA	BB	BC	BD	BE	BF
ADA	C0	C1	C2	C3	C4	C5	C6	C7
SDA	C8	C9	CA	CB	CC	CD	CE	CF
MPY	D0	D1	D2	D3	D4	D5	D6	D7
DIV	D8	D9	DA	DB	DC	DD	DE	DF
EST	E0	E1	E2	E3	E4	E5	E6	E7
SST	E8*	E9*	EA*	EB*	EC*	ED*	EE*	EF*
ZJP	F0	F1	F2	F3	F4	F5	F6	F7
CJP	F8	F9	FA	FB	FC	FD	FE	FF

NOT USED * SPECIAL MEANING

Westinghouse Computer Department • Orlando, Florida